Food Ordering via QR Codes in Restaurant

Submitted by

MD. ABU AHOSAN HABIB

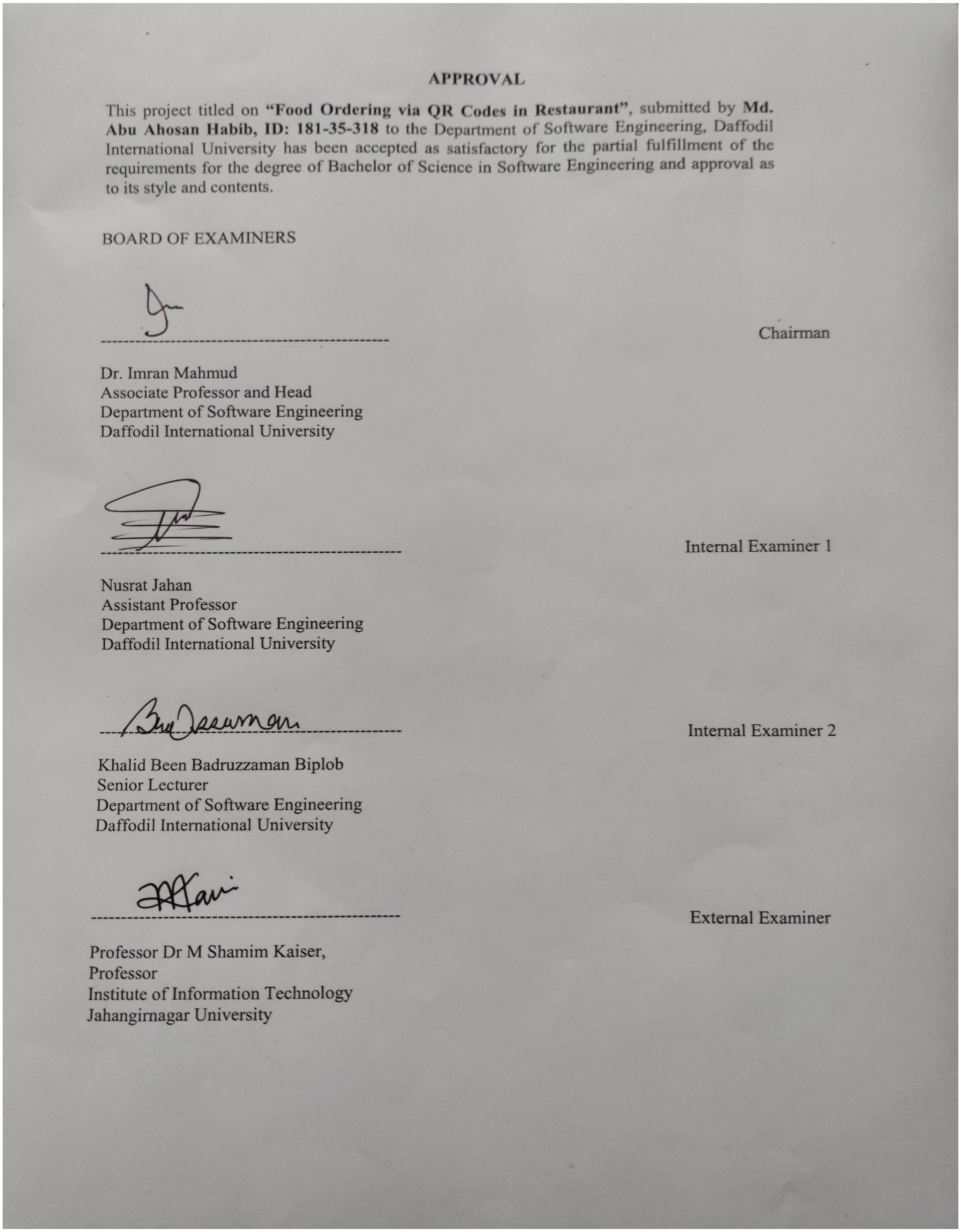
ID: 181-35-318
Department of Software Engineering
Daffodil International University

Supervised by

Mr. Khalid Been Md.
Badruzzaman Biplob
Lecturer (Senior Scale)
Department of Software Engineering
Faculty of Science and Information Technology
Daffodil International University

This project report has been submitted in fulfillment of the requirements for the degree of BSc in Software Engineering.

© All right Reserved by Daffodil International University



APPROVAL

This project titled on **"Food Ordering via QR Codes in Restaurant"**, submitted by **Md. Abu Ahosan Habib, ID: 181-35-318** to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Software Engineering and approval as to its style and contents.

BOARD OF EXAMINERS

Chairman

Dr. Imran Mahmud
Associate Professor and Head
Department of Software Engineering
Daffodil International University

Internal Examiner 1

Nusrat Jahan
Assistant Professor
Department of Software Engineering
Daffodil International University

Internal Examiner 2

Khalid Been Badruzzaman Biplob
Senior Lecturer
Department of Software Engineering
Daffodil International University

External Examiner

Professor Dr M Shamim Kaiser,
Professor
Institute of Information Technology
Jahangirnagar University

Fall-2021
DAFFODIL INTERNATIONAL UNIVERSITY

Declaration of Project:

Author Full Name : Md. Abu Ahosan Habib
Date of Birth : 10-10-2000
Title of Project : Food Ordering via QR Codes in Restaurant
Academic Session : 2018-2021

I, the author, hereby declare that the project becomes the property of Daffodil International University (DIU) and to be placed at the department of Software Engineering. I also give permission to the DIU to reproduce this project in whole or in part for the purpose of research or academic exchange only.

Certified by:

Ahosan Habib

SIGNATURE

SIGNATURE OF SUPERVISOR

Md. Abu Ahosan Habib(181-35-318)
STUDENT NAME & STUDENT ID

Date:30-12-2021

Mr. Khalid Been Md. Badruzzaman Biplob
NAME OF SUPERVISOR

Date: 30-12-2021

ACKNOWLEDGEMENT

In this present world of competition there is race of existence in which those are having will come to forward succeed. Project is a bridge between theoretical and practical working. With this willing I joined this particular project. First, I would like to thank the supreme power the Almighty ALLAH who is obviously the one who is guided me to work on the right path of life. Without his grace this project could not become a reality. Next to him are my parents, whom I am greatly indebted for brought up with love and encouragement to this stage.

I am feeling oblige in talking the opportunity to study in Daffodil International University. I would like to sincerely thank to Associate Professor Dr. Imran Mahmud, Head, Department of Software Engineering. All the honorable teachers who teach me such an interesting and understandable way full of enjoyment. I am grateful to having them in my journey.

I am exceptionally obligated to Daffodil International University for their direction and steady supervision by Mr. Khalid Been Md. Badruzzaman Biplob and in addition for giving necessary information with respect to the venture and additionally for their help in finishing the project.

I am grateful to my Department staff members, Lab technicians and Non-teaching staff members for their extreme help throughout my project.

Finally, I would like to express my love to my batch mate, member of DIU for their kind co-operation and consolation which help me in finishing of this task.

ABSTRACT

For a restaurant, food ordering is a critical service. This is a service that a waiter provides to a customer in the restaurant. There are various issues that may arise if traditional food ordering is used. Misunderstandings between the server and the customer when taking the order are common problems. Furthermore, the consumer must wait for a waiter to arrive and receive the order. The existing system takes orders from clients in the traditional manner, utilizing a piece of paper and menu paper. As a result, the Food Ordering System with QR Code Technology is a real-time ordering system for managing the restaurant's order process. As a result, using QR Code technology to create a food ordering system is an alternative to solving that problem. Because smartphones are now a necessity for everyone, the system uses them as a platform. Customers must scan the QR Code on the paper provided at each table by the restaurant. The consumer can also confirm the ordered item using this technique. The restaurant's personnel can also manage the menu and view the order list.

Table of Contents

CHAPTER 3: SYSTEM ANALYSIS

Chapter 4: System Design Specification

List of Tables

List of Figures

List of Abbreviations

DIU = Daffodil International University

SWE = Software Engineering Department

CHAPTER-01

Introduction

1.1 Project Overview

The ”Food Ordering via QR Codes in Restaurant” Project is an entirely new system to handle the ordering systems inside a restaurants. This project offers an efficient ordering system for the restaurants. In this project, the customer scans a QR code on his / her mobile, sees the menu and places the order via the mobile without downloading any app. Customer can payment for order using card/mobile banking or also payment in cash. After confirmation order, Kitchen cooking food and ready for serve. At last waiter can serve the food.

1.2 Background

Firstly, We know, The waiter serves as a link between the customer and the cooking personnel in the department. Waiters are prone to making human errors, such as misunderstandings with customers. This misunderstanding will have an impact on the food preparation procedure. In this situation, the consumer will be dissatisfied if the food requested does not match the food given to them.

Secondly, the customer may encounter the issue of having to wait a lengthy time for the waiter to arrive and take their order. This difficulty can be handled with this system because customers can place their own orders without having to wait for a waiter.

Finally, in most restaurants, orders are taken on paper, which is then passed to the kitchen department. This may result in the misplacement of the ordered paper, requiring the waiter to solicit a new order from the customers. This issue will be solved if the kitchen has an application that allows them to view the ordered menu in a systematic manner without having to use paper.

1.3 Objective

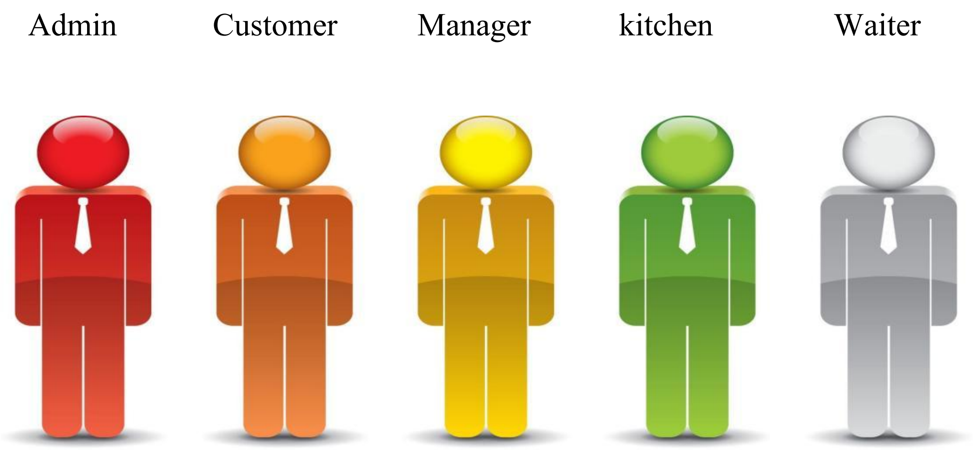
The goal of this project is to create a QR code-based food ordering system. The following objectives are listed in order to fulfill the project's goal:

- To create a food ordering system based on QR codes in order to avoid any ordering errors.
- It's saves time. Because For ordering food, Don’t need wait for waiters.
- This system helps to reduce the restaurant cost.

1.4 Scopes

- This system can be used to the any restaurant.
- The users of this system are divided into two categories:
  - i. The restaurant's clients, who will place an order for their food.
  - ii. The restaurant’s employees, who will manage this software.

1.5 Stakeholders



Primary Stakeholder	Customer, Manager, kitchen, Waiter
Secondary Stakeholder	Admin

1.6 Project Schedule

A timetable is a list of project milestones, activities, and deliverables, usually with expected start and conclusion dates, in project management. It specifies when the project began and when it will be completed, as well as how many times each section of the project model has been used and the release date. A schedule is often used in project planning and portfolio management.

1.6.1 Gantt Chart

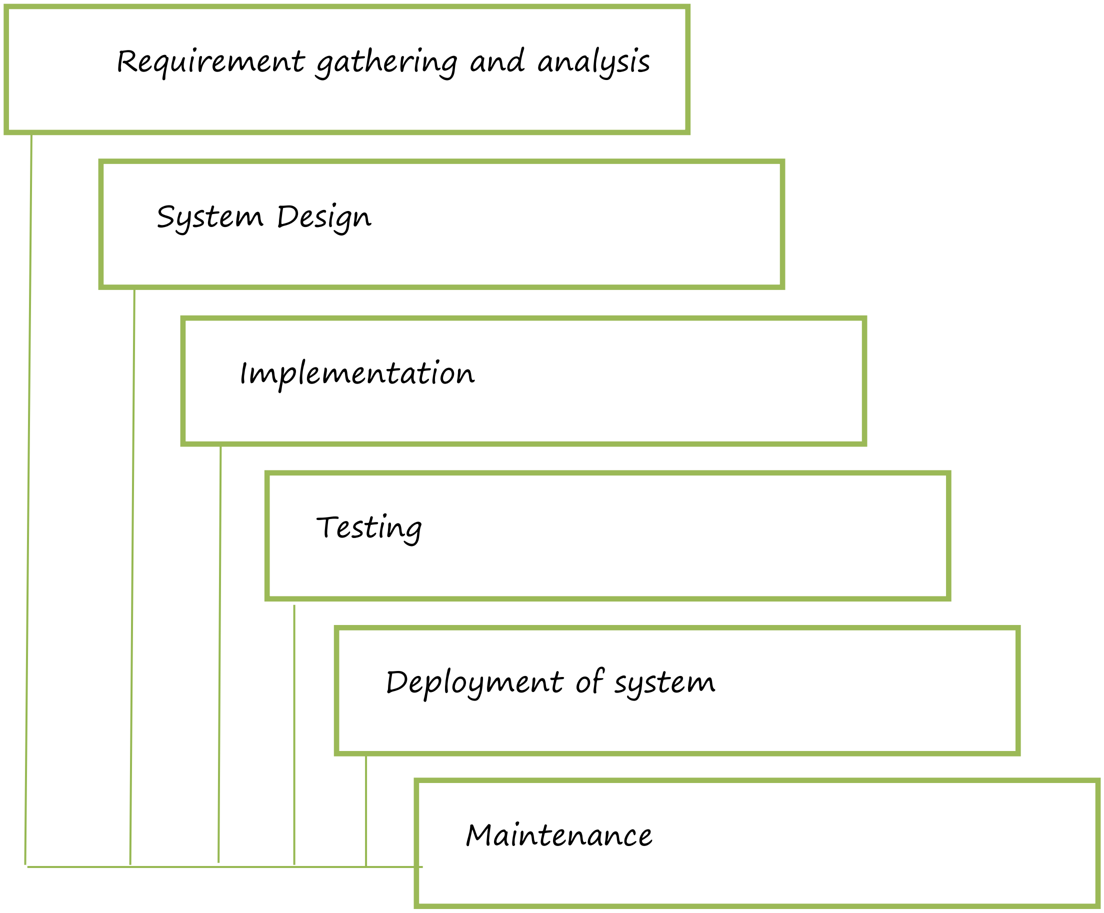
Weeks→ **Works↓**	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Requirement Gather	🟫	🟫																
Analysis			🟦															
Feasibility Study				⬛	⬛													
Project Proposal						🟦												
UML Diagrams							⬛											
Project UI								🟧	🟧	🟧	🟧	🟧	🟧	🟧				
Mid-term defense												🟥						
Implementation									🟨	🟨	🟨	🟨	🟨	🟨	🟨			
Testing											🟨	🟨	🟨	🟨	🟨			
Documentation																🟥	🟥	
Final Defense																		🟩

Figure 1.6.1: Gantt Chart

1.7 Methodology to be used for the project

The methodology to be used is Agile Method. This is because the system will be standing alone with a new prototype within very short span. It will therefore increase the chances of the system to be reviewed by the user of the system at the end of each prototype to make any amendments necessary until the final prototype is released. This will help the system to be developed more efficiently with the user requirements fully met.

Here is the developing methodology we are following-



CHAPTER-02
Software Requirement Specification

2.1 Functional Requirements

The functional requirements referred to a mandatory function which mandatory to the system. It must be able to perform for the web and also all kinds of software systems. Here I discuss about my project's functional requirements.

Table 2.1: Functional Requirements

FR-01	**Log In**
Description	Authentication of user whenever he/she logs into the system.
Functional / Non functional	Functional
Stakeholders	Admin, Kitchen, Manager, Waiter
Priority	High

FR-02	**Manager Management**
Description	This module helps admin to Add,Update & Delete manager information. Admin is able to maintain all the information of manager.
Functional / Non functional	Functional
Stakeholders	Admin
Priority	High

FR-03	**Kitchen Management**
Description	This module helps admin to Add,Update & Delete kitchen information. Admin is able to maintain all the information of kitchen.
Functional / Non functional	Functional
Stakeholders	Admin
Priority	High

FR-04	**Waiter Management**
Description	This module helps admin to Add,Update & Delete Waiter information. Admin is able to maintain all the information of Waiter .
Functional / Non functional	Functional
Stakeholders	Admin
Priority	Medium

FR-05	**QR Code Generate**
Description	This module helps admin to Generate a new QR code using table number.
Functional / Non functional	Functional
Stakeholders	Admin
Priority	High

FR-06	**QR Code Scan**
Description	Customer scan qr code by his/her smartphone. Qr code provided every table.
Functional / Non functional	Functional
Stakeholders	Customer
Priority	High

FR-07	**Food Menu**
Description	When a customer scan a qr code then go to restaurant website and showing food menu his/her phone display.
Functional / Non functional	Functional
Stakeholders	Customer
Priority	High

FR-08	**Menu Management**
Description	This module helps kitchento Add,Update & Delete menu information. Kitchen is able to maintain all the information of Menu.
Functional / Non functional	Functional
Stakeholders	kitchen
Priority	Medium

FR-09	Select Food from Menu
Description	This module helps customer to choose food for order.
Functional / Non functional	Functional
Stakeholders	Customer
Priority	High

FR-10	Cart management
Description	In this module customer can select food quantity,remove food and also see Total amount.
Functional / Non functional	Functional
Stakeholders	Customer
Priority	High

FR-11	Checkout from cart
Description	In this module customer can placed his/her order after checking.
Functional / Non functional	Functional
Stakeholders	Customer
Priority	High

FR-12	Review order
Description	This module helps to customer for overview his/her final order before payment.
Functional / Non functional	Functional
Stakeholders	Customer
Priority	Medium

FR-13	Payment of Order
Description	This module helps customer to payment ordered food.customer choose payment option cash/card/mobile banking.
Functional / Non functional	Functional
Stakeholders	Customer
Priority	High

FR-14	Confirm the order
Description	After payment successful then confirm the order.
Functional / Non functional	Functional
Stakeholders	Customer, Manager
Priority	High

FR-15	After payment Order Details
Description	This module helps to overview final order after payment.
Functional / Non functional	Functional
Stakeholders	Customer, Manager, Kitchen
Priority	Medium

FR-16	Order Status
Description	This module helps to customer check the order status. Kitchen update order status is food ready and waiter update order status is food served.
Functional / Non functional	Functional
Stakeholders	Customer, Waiter, Kitchen
Priority	High

2.2 Non-Functional Requirement

Here I discuss about my website non-functional requirements.

Table 2.2: Non-Functional Requirements

ID	Name	Description	Non-Functional Requirement	Priority
NFR-01	Security	Using token-based authentication, session, validation 2FA it will be secure from unauthorized access.	Non-Functional	High
NFR-02	Availability	The system should work 24/7 as user can get access and service.	Non-Functional	High
NFR-03	Accuracy	Data or process requirement concerned with defining the precision which the solution will record or produce data.	Non-Functional	High
NFR-04	Maintenance	It's way how easy to support, change and enhance the system.	Non-Functional	Medium

2.3 Performance Requirements

2.3.1 Speed and Latency Requirements

1. Dataset would be inserted in MySQL Databases using php laravel.
2. UI design-build on the user fulfills table data set and show. The Database we have to use Mysql.

2.3.2 Precisions or Accuracy Requirements

For all types of users, it must be needed to accurate. Wrong information might be ruined the system process.

1. All users are capable to show accurate page. He or she can make order and kitchen can see their order
2. Kitchen Manage food menu which are available or not.
3. Admin can update user's information.

2.3.3 Capacity Requirements

We must develop a system which is capable to handle user, provide accurate information, handling database, manage HTTP request, etc.

1. At time multiple users can use this system.
2. The system needs to handle thousands of data every month.

2.4 Dependability Requirements

2.4.1 Reliability Requirements

1. Admin, Manager, kitchen, Waiter should be log in to the system using his or her valid email and password.
2. Admin can easily update Manager, kitchen, Waiter information.
3. Customer scan qr code and order the food.
4. Kitchen can see the order and prepare the food.
5. Customer track his/her order status.

2.4.2 Availability Requirements

1. When we are using this system need to run apace server.
2. Need to know command for run properly and easily.
3. This is a web application should run on a web browser. (Preferable browser is Chrome, Firefox, Internet Explorer).

2.5 Maintainability and Supportability Requirements

2.5.1. Maintenance Requirements

1. Web application did not modify or change.
2. If we need to data need to recover or update then need to using command line.
3. We are need to maintainers all security and others works.

2.5.2 Supportability Requirements

1. When the system did not work perfectly then need to check database and others work.
2. Update security patch and others system.

2.5.3 Adaptability Requirements

1. This is very important website.
2. This website can help to customer and restaurant, they can get all services easily.

2.5.4 Scalability or Extensibility Requirements

1. This website is good and reliable for each and every one.
2. Website is good for help.
3. Every customer can order food easily.

2.6 Security Requirements

2.6.1 Access Requirements

1. Only registered user can access admin panel .
2. Customer can order without registration.

2.6.2 Privacy Requirements

1. Here we are using activate and non-activate account system
2. All customer account info hidden from others.

2.7 Usability and Human Interaction Requirements

This requirement defines how to meet the physical needs of the intended users of our website.

2.7.1 Ease of Use Requirements

The system is easy to use and can easily be understood.

UH-01	The system must be usable for all associate stakeholders
Description	The system indicates the several possible that the users.
Stakeholders	System Admin, Customer, Manager, Kitchen, Waiter.

2.7.2 Understand Ability and Politeness Requirements

This section describes more requirements.

UH-02	The features of the Food Ordering via QR Codes in Restaurant
Description	The system can more efficiently ease of use more added features. The system is understanding ability for both users. The system not use any term that is not specified in this website.
Stakeholders	System Admin

2.7.3 Accessibility Requirements

There are no access requirements beside those that has been outline in the

below:

AR-1: Log in as a Admin
AR-2: Log in as a Manager
AR-3: Log in as a Kitchen
AR-4: Log in as a Waiter
AR-5: Log out as Admin
AR-6: Log out as Manager
AR-7: Log out as Kitchen
AR-8: Log out as a Waiter

To get access to this system or a specific module the system must provide a control. In order to prevent anyone to exploit stolen all user's passwords must be encrypted in the hash process

2.7.4 User Documentation Requirements

UH-03	The system developer documentation
Description	Develop this project I have specified requirements of user's documentation.
Stakeholders	System Developer

2.8 Look and Feel Requirements

Look and feel requirements mainly refer to how the system will look and how the user interface or graphical user interface of our system will display to the end-users.

2.8.1 Appearance Requirements

Admin and customer must know which input fields are required and which are not. For that reason, we will use labels for all input fields. Input fields might be text type, radio, checkbox, spinner, etc.

LF-01	Labels of mandatory fields must be bold.
Description	Label of obligation fields must be bold to identify them as being of obligation.
Stakeholders	Admin, Customer.

2.8.2 Style Requirements

We will provide a website user interface. This requirement does not only define the necessity to use a css but although the requirements regarding's the css's content as well as css framework like bootstrap.

LF-02	The look and feel must be controllable using a style sheets.
Description	The styling of the elements of the web application user interface will be define using css, JS, Bootstrap.
Stakeholders	System Admin

2.9 Operational and Environmental Requirements

This requirement focuses on how the users operate the system, including interfaces and interoperability with other systems. The requirement established how good and under what condition the system must perform.

2.9.1 Expected Physical Requirements

There are no specific expected physical requirements.

2.9.2 The Requirements for Interfacing with Adjacent System

There is no specific interface with adjacent system requirements.

2.9.3 Release Requirements

There is no specific release requirement but in the project schedule. It was described briefly.

2.10 Legal Requirements

These requirements consider any violence of rules and regulation and which rules should be followers to maintainers these systems.

2.10.1 Compliance Requirements

There are no specific compliance requirements.

2.10.2 Standard Requirements

There are no specific standard requirements.

CHAPTER-03

SYSTEM ANALYSIS

3.1 Use case Diagram

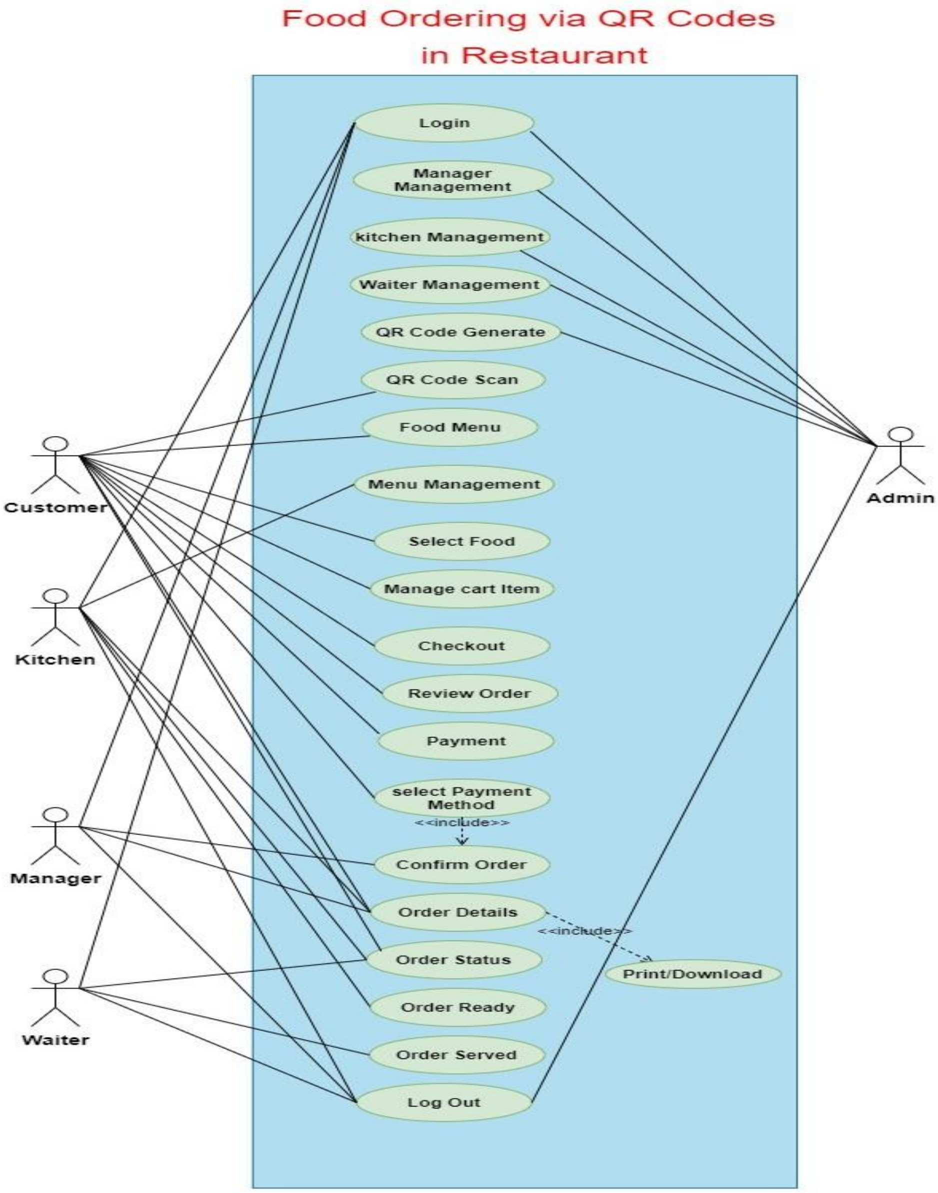


Figure 3.1: Use Case

3.2 Use Case Description

3.2.1 Login and Log out System

Use Case	Login and log out system	
Goal <a longer statement of the goal in context if needed>	Admin, Manager, Kitchen, Waiter can access the system. And finally, Log out.	
Preconditions <what we expect is already the state of the world's>	N/A	
Success End Condition <the state of the world upon successfully complete>	Admin, Manager, Kitchen, Waiter can access this system	
Failed End Condition <the state of the world if goal abandoned>	Admin, Manager, Kitchen, Waiter can access the system	
Primary Actors:	Admin, Manager, Kitchen, Waiter	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Access This System	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1	Admin, Manager, Kitchen, Waiter
	2	The user enters his or her email id and password in the returning user section of the sign in screen.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1a	The user enters his or her username and password
Quality Requirements	**Step**	**Requirement**
	1	When user login then needs to correct email id for login and password.

3.2.2 Staff management

Use Case	Staff Management(manager,kitchen,waiter).	
Goal <a longer statement of the goal in context if needed>	Admin can add,update,delete manager,kitchen and waiter information.	
Preconditions <what we expect is already the state of the worlds>	Must be logged in to the system.	
Success End Condition <the state of the world upon successfully complete>	System get manage staff request from admin	
Failed End Condition <the state of the world if goal abandoned>	System cannot get manage staff request from admin.	
Primary Actors:	Admin	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Access This System.	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.	Admin can manage staff.
	2.	Admin click manage staff then the URL show the "manage staff." pages.
	3.	Admin enter all required field and click manage staff button.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1.	Get "manage staff." page again load if any error occurred.
Quality Requirements	**Step**	**Requirement**
	1.	Must be logged in to the system

3.2.3 Scan Qr Code

Use Case	Scan Qr Code	
Goal <a longer statement of the goal in context if needed>	Customer can scan qr code for get the menu and order food.	
Preconditions <what we expect is already the state of the worlds>	Must be have qr code scanner.	
Success End Condition <the state of the world upon successfully complete>	System get restaurant page request by customer	
Failed End Condition <the state of the world if goal abandoned>	Cannot get.	
Primary Actors:	Customer	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Qr code.	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.	The customer can open qr code scanner.
	2.	Customer scan qr code from table.
	3.	Customer open qr code scanner andscan qr code from table.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1.	Qr code does not match then show try again .
Quality Requirements	**Step**	**Requirement**
	1.	Qr code needed.

3.2.4 Select Food

Use Case	Select Food.	
Goal <a longer statement of the goal in context if needed>	Customer can Select food.	
Preconditions <what we expect is already the state of the worlds>	N/A	
Success End Condition <the state of the world upon successfully complete>	System get Select food request from customer	
Failed End Condition <the state of the world if goal abandoned>	System cannot get elect food request from admin.	
Primary Actors:	Customer	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Access This System.	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.	Customer can select food.
	2.	Admin click menu then the URL show the “select food” pages.
	3.	Admin enter all required field and click menu button.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1.	Get “menu” page again load if any error occurred.
Quality Requirements	**Step**	**Requirement**
	1.	Must be scan qr code.

3.2.5 Menu Management

Use Case	Menu Management(manager,kitchen,waiter).	
Goal <a longer statement of the goal in context if needed>	Kitchen can add,update,delete menu information.	
Preconditions <what we expect is already the state of the worlds>	Must be logged in to the system.	
Success End Condition <the state of the world upon successfully complete>	System get manage menu request from kitchen	
Failed End Condition <the state of the world if goal abandoned>	System cannot get manage menu request from kitchen.	
Primary Actors:	kitchen	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Access This System.	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.	Kitchen can manage menu.
	2.	Kitchen click menu management then the URL show the "menu management." pages.
	3.	Kitchen enter all required field and click manage menu button.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1.	Get "menu management" page again load if any error occurred.
Quality Requirements	**Step**	**Requirement**
	1.	Must be logged in to the system

3.2.6 Update Order Cart

Use Case	Update order cart	
Goal <a longer statement of the goal in context if needed>	User can update order cart.	
Preconditions <what we expect is already the state of the worlds>	Must be select food.	
Success End Condition <the state of the world upon successfully complete>	System get update cart info from the user.	
Failed End Condition <the state of the world if goal abandoned>	System don't get update cart info from the user.	
Primary Actors:	User	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Update orderCart	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.	User can update order cart.
	2.	User enter the URL & the URL show the "Order cart" pages.
	3.	User can delete or add food quantity.
	4.	System show update quantity and price.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1.	Get "Order cart" page again load if any error occurred.
Quality Requirements	**Step**	**Requirement**
	1.	Must be scan qr code.

3.2.7 Select Payment

Use Case	Select Payment.	
Goal <a longer statement of the goal in context if needed>	User can select payment upon place order.	
Preconditions <what we expect is already the state of the worlds>	Must be add food to cart.	
Success End Condition <the state of the world upon successfully complete>	System get select payment info request from user.	
Failed End Condition <the state of the world if goal abandoned>	System cannot get select payment info request from user.	
Primary Actors:	User	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Access This System.	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.	User can select payment upon place order.
	2.	Users enter the URL & the URL show the "Checkout" pages.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1.	Get "Checkout" page again load if any error occurred.
Quality Requirements	**Step**	**Requirement**
	1.	Must be select the food.

3.2.8 Check Out

Use Case	Check Out.	
Goal <a longer statement of the goal in context if needed>	User can place order after select payment.	
Preconditions <what we expect is already the state of the worlds>	Must be select payment option	
Success End Condition <the state of the world upon successfully complete>	System get order info request from user.	
Failed End Condition <the state of the world if goal abandoned>	System cannot get order info request from user.	
Primary Actors:	User	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Access This System.	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.	User can place order after select payment..
	2.	Users enter the URL & the URL show the “Checkout” pages.
	3.	User click confirm button and system show order confirmation.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1.	Get “Checkout” page again load if any error occurred.
Quality Requirements	**Step**	**Requirement**
	1.	Must be select the food and select payment.

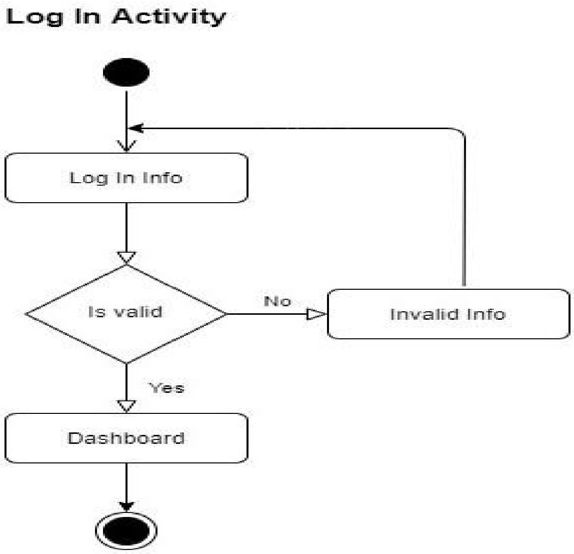
3.2.9 Track Order

Use Case	Track Order.	
Goal <a longer statement of the goal in context if needed>	User can track order after checkout.	
Preconditions <what we expect is already the state of the worlds>	Must be ordered food from this site.	
Success End Condition <the state of the world upon successfully complete>	System get order id request from user.	
Failed End Condition <the state of the world if goal abandoned>	System cannot get order id request from user.	
Primary Actors:	User	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Access This System.	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.	User can track order after checkout.
	2.	Users enter the URL & the URL show the “Track Order” pages.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1.	Get “Track Order” page again load if any error occurred.
Quality Requirements	**Step**	**Requirement**
	1.	Must be ordered food from this site.

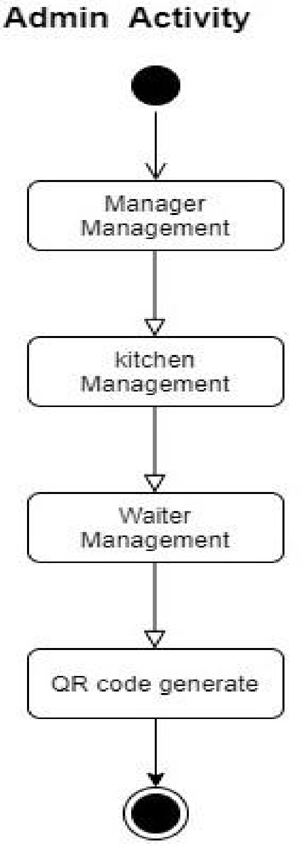
3.2.10 Update Order Status

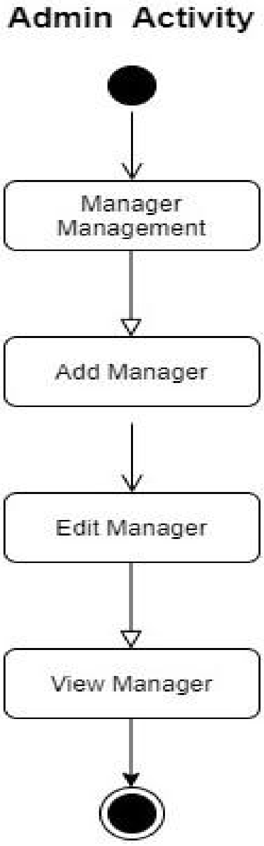
Use Case	Update order status.	
Goal <a longer statement of the goal in context if needed>	Kitchen can update order status.	
Preconditions <what we expect is already the state of the worlds>	Must be logged in to the system.	
Success End Condition <the state of the world upon successfully complete>	System get update order info request from kitchen	
Failed End Condition <the state of the world if goal abandoned>	System cannot get update order info request from kitchen.	
Primary Actors:	Kitchen	
Secondary Actors	N/A	
Trigger <the action upon the system that starts use case>	Access This System.	
Description/main Success Scenario <the steps of the scenario from the trigger to goal delivery and any clean up after>	**Step**	**Action**
	1.	Kitchencan update order status.
	2.	Kitchen click ordered food then the URL show the "order." pages.
	3.	Kitchen update order info and click save button.
Alternative Flows <a: condition causing branching> <a1: action or name of sub use case >	**Step**	**Branching Action**
	1.	Get "update order info" page again load if any error occurred.
Quality Requirements	**Step**	**Requirement**
	1.	Must be logged in to the system

3.3 Activity Diagram

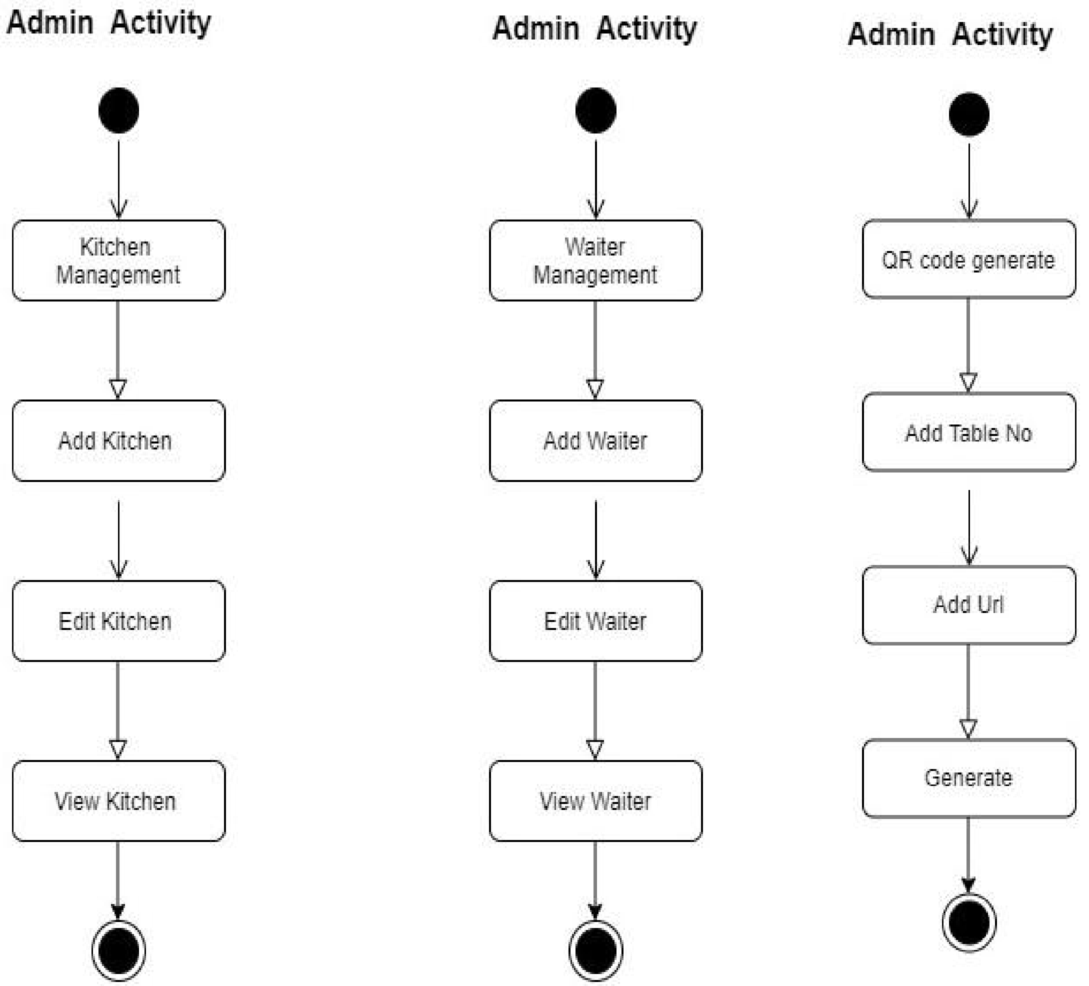


3.3.1 Admin Activity Diagram



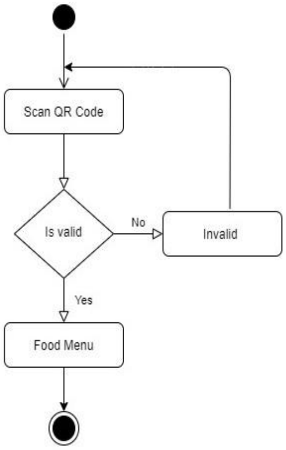


Admin

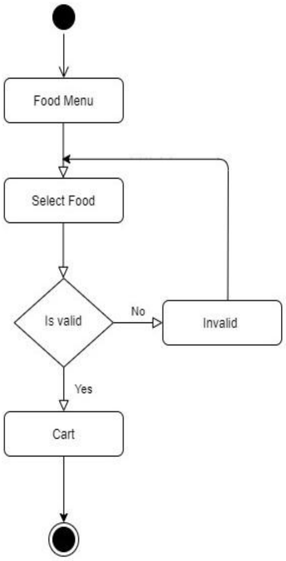

Admin Activity
Kitchen Management
Add Kitchen
Edit Kitchen
View Kitchen
Admin Activity
Waiter Management
Add Waiter
Edit Waiter
View Waiter
Admin Activity
QR code generate
Add Table No
Add Url
Generate

3.3.2 Customer Activity Diagram

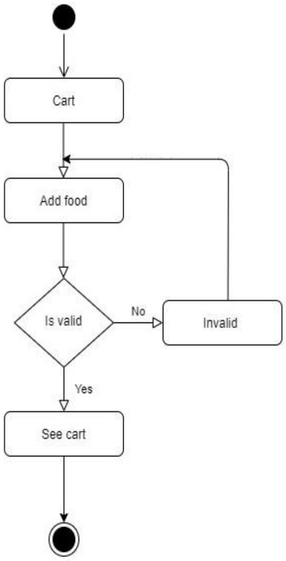
Customer Activity



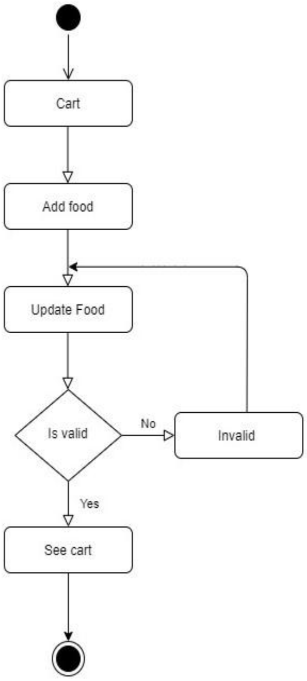
Customer Activity



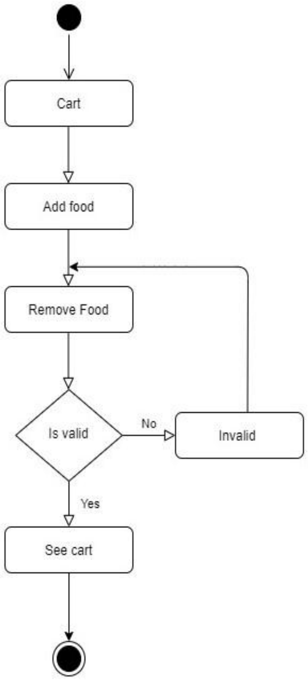
Customer Activity
Add to Cart



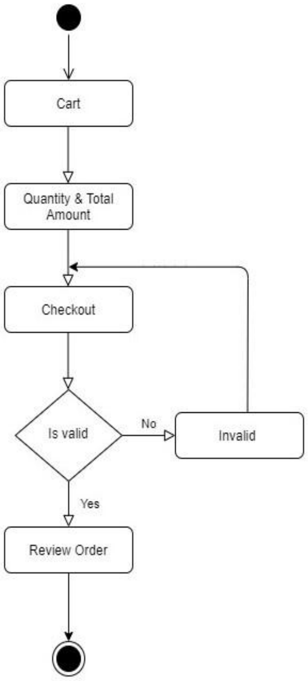
Customer Activity
Update from Cart



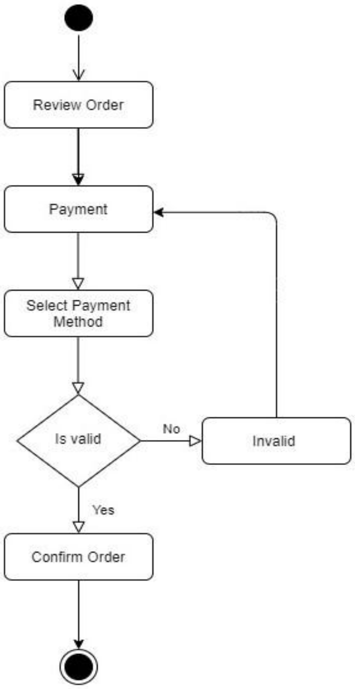
Customer Activity
Removed from Cart



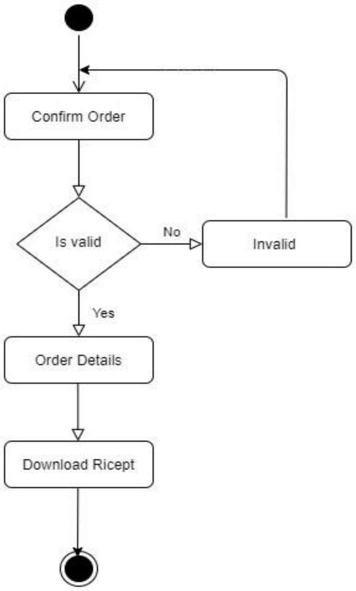
Customer Activity
Checkout Order



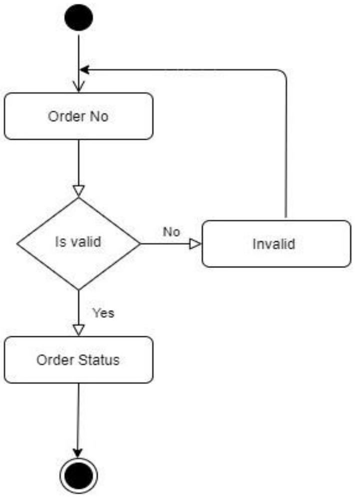
Customer Activity
Payment



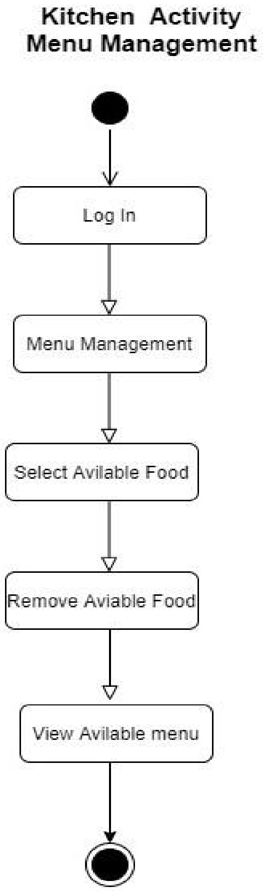
Customer Activity
Order Details

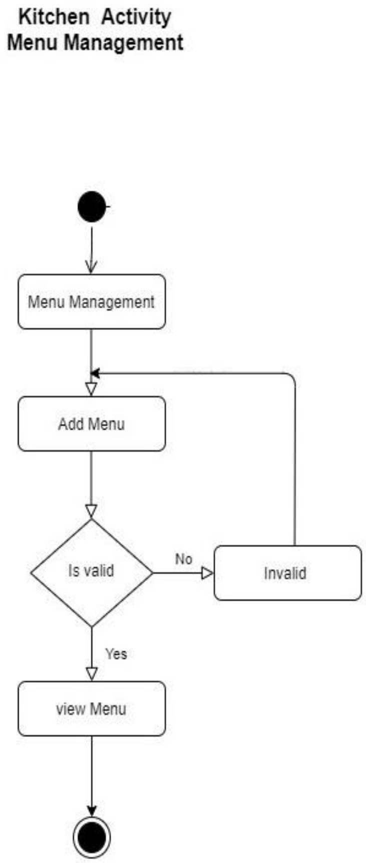


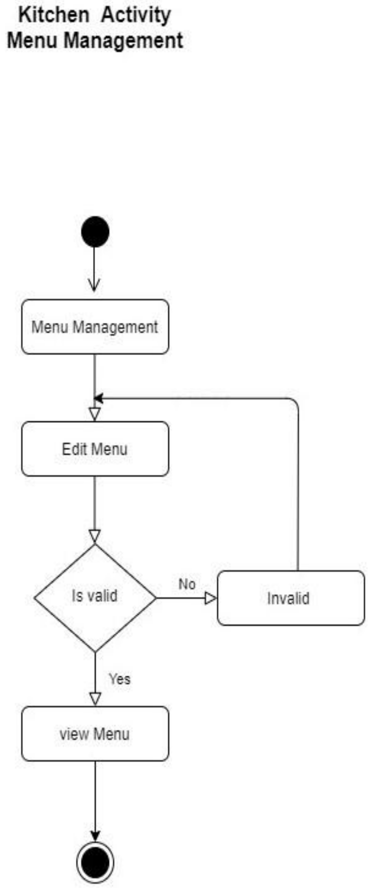
Customer Activity
Order Status

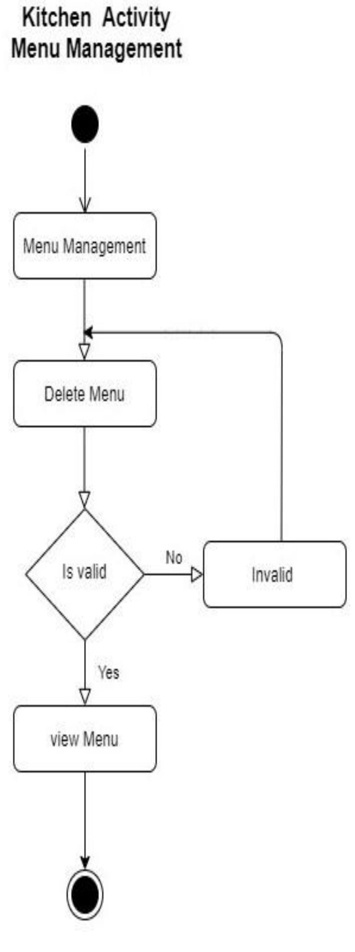


3.3.3 Kitchen Activity Diagram

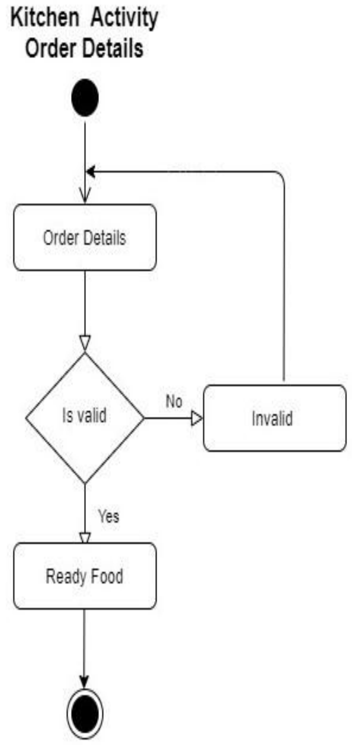




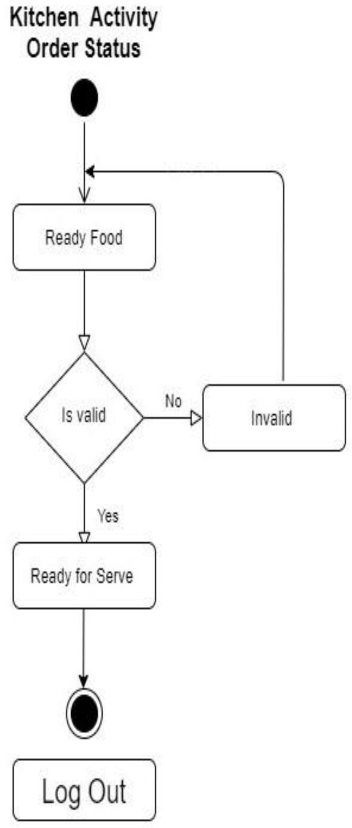




Kitchen Activity
Menu Management



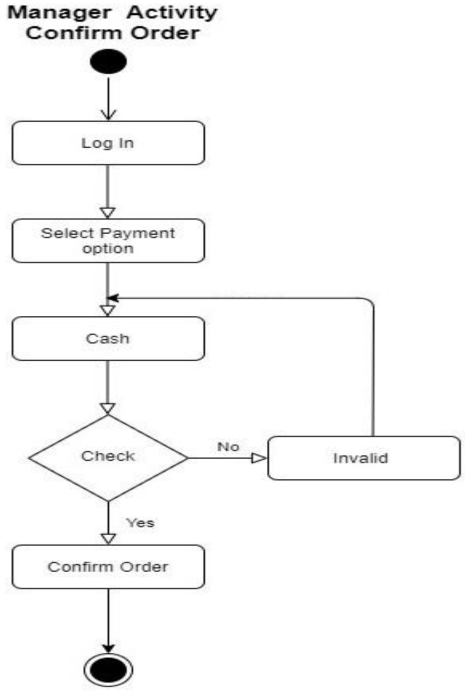
Kitchen Activity
Order Details

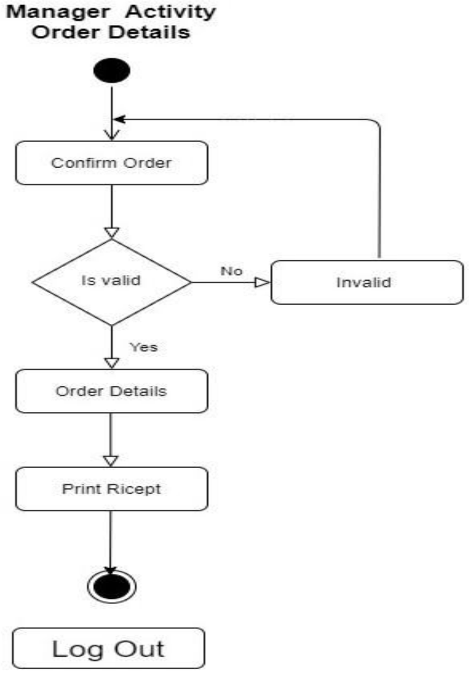


Kitchen Activity
Order Status

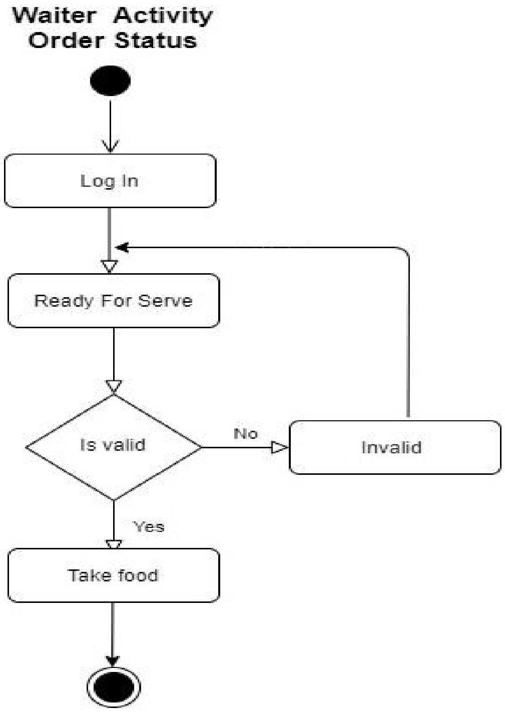
© Daffodil International University

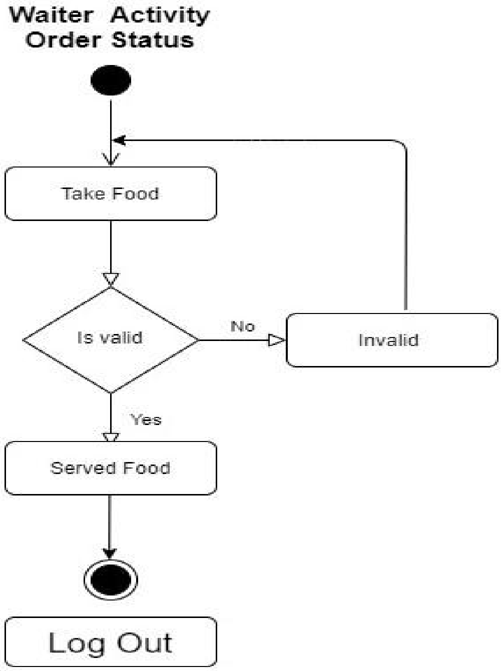
3.3.4 Manager Activity Diagram





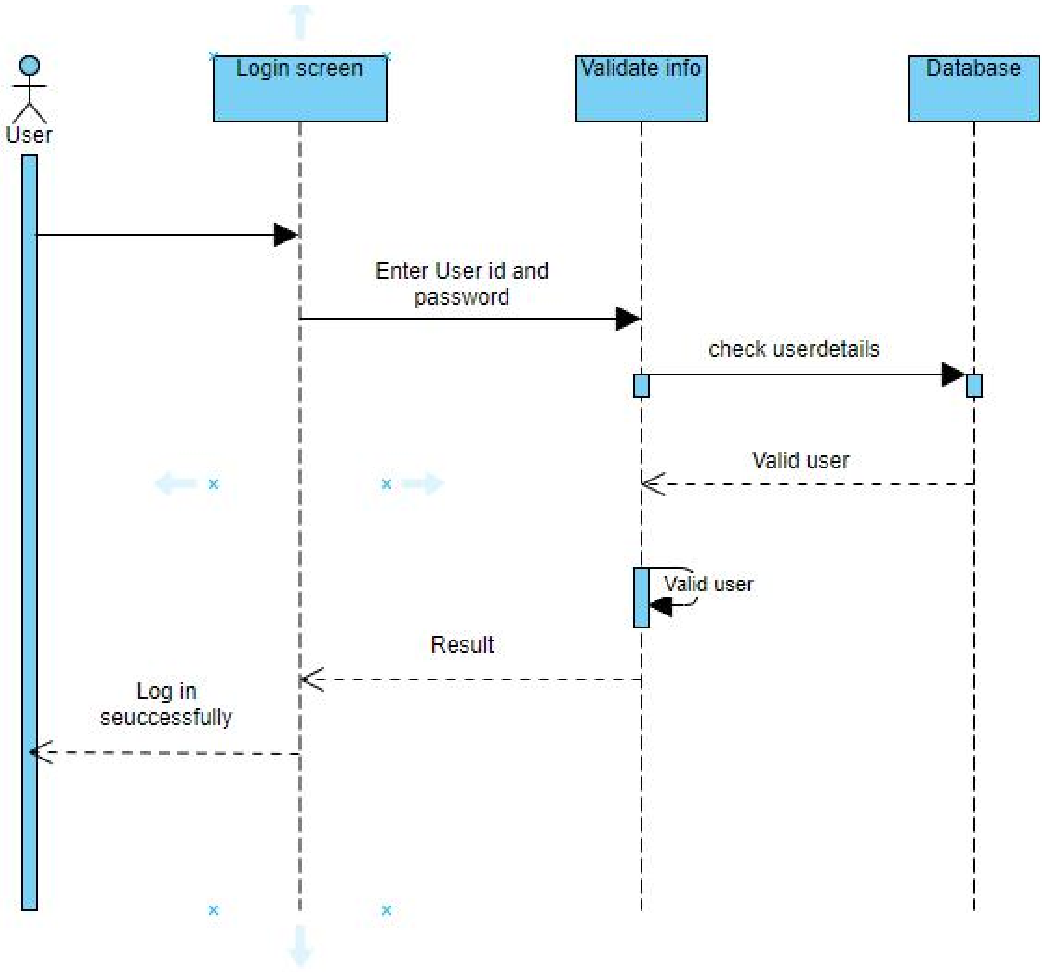
3.3.5 Waiter Activity Diagram





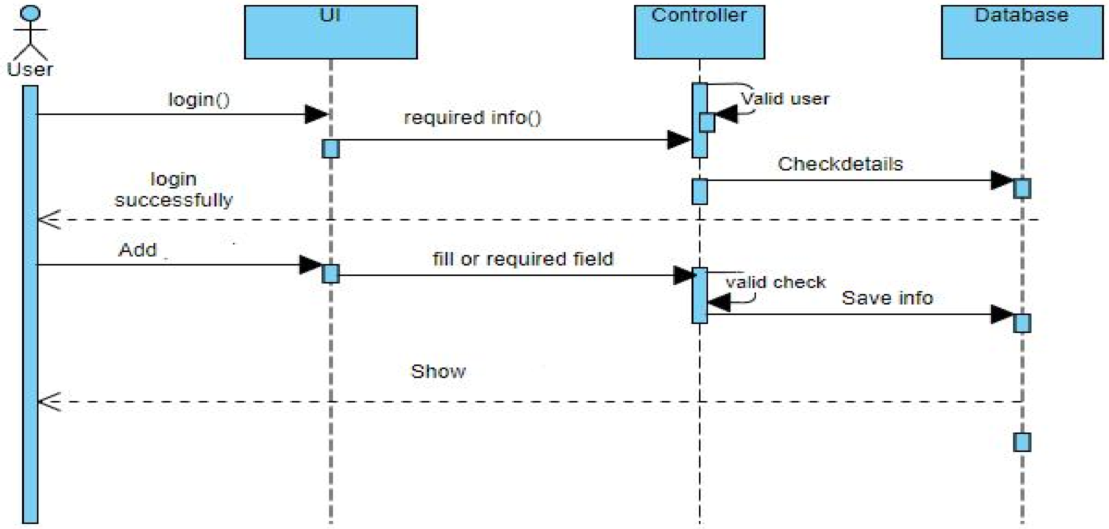
3.4 System Sequence Diagram

3.4.1 User login sequence

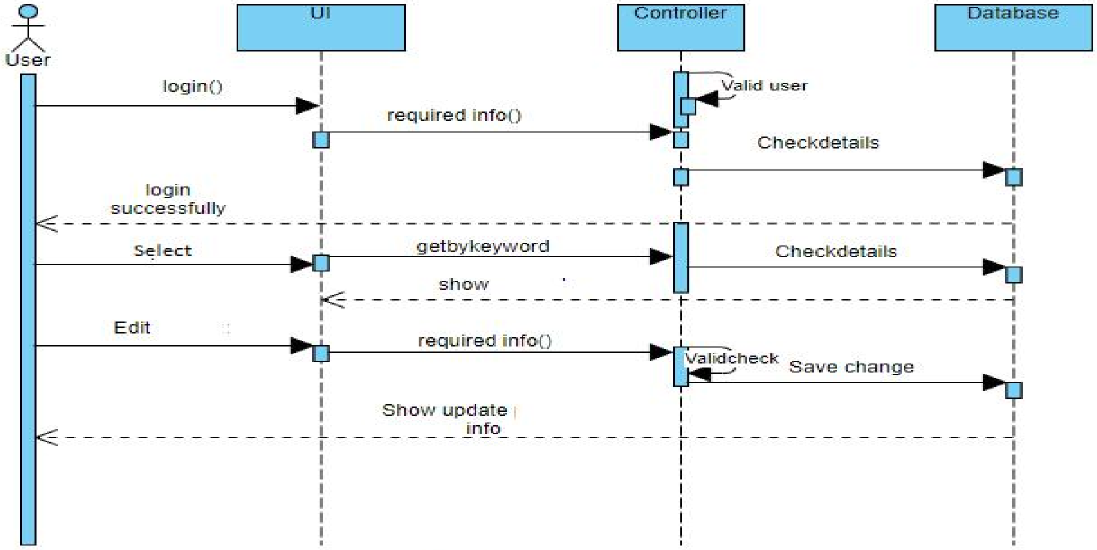


© Daffodil International University

3.4.2 Menu, Manager, Kitchen, Waiter information add and update

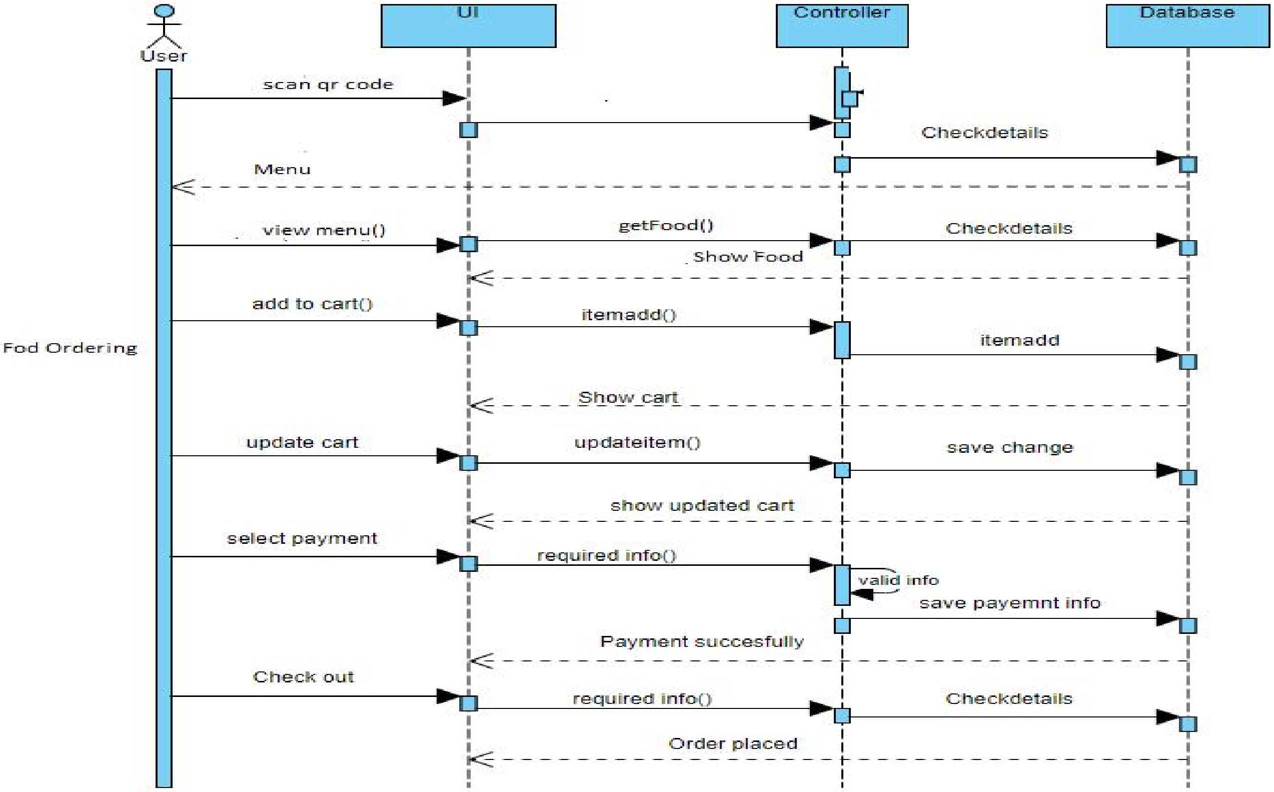


Add

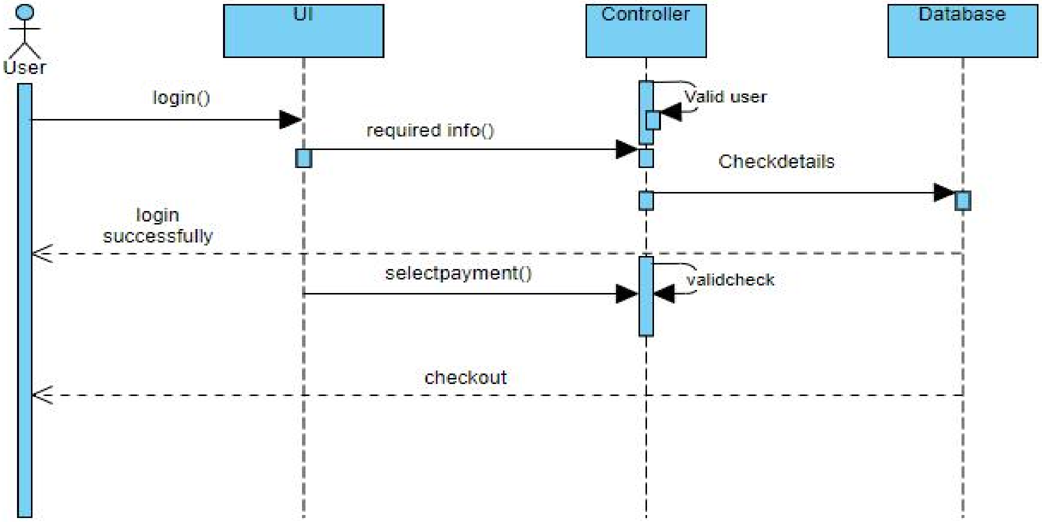


Edit Info

3.4.3 Customer Food Ordering

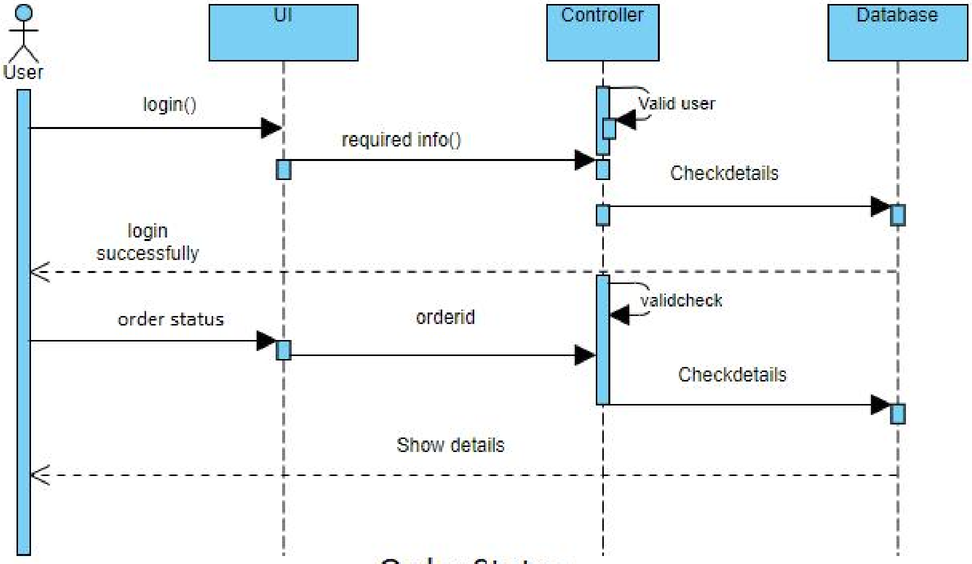


3.4.4 Select Payment



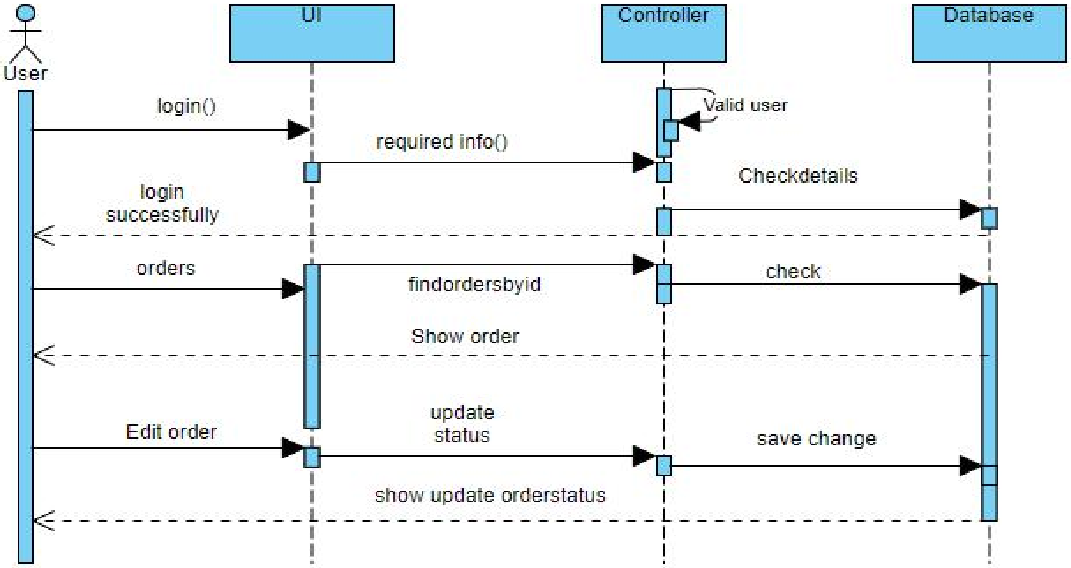
Select Payemt

3.4.5 Order Status



Order Status

3.4.6 Update Order Status



Update order status

CHAPTER-04

System Design Specification

4.1 Entity Relationship Diagram (ERD):

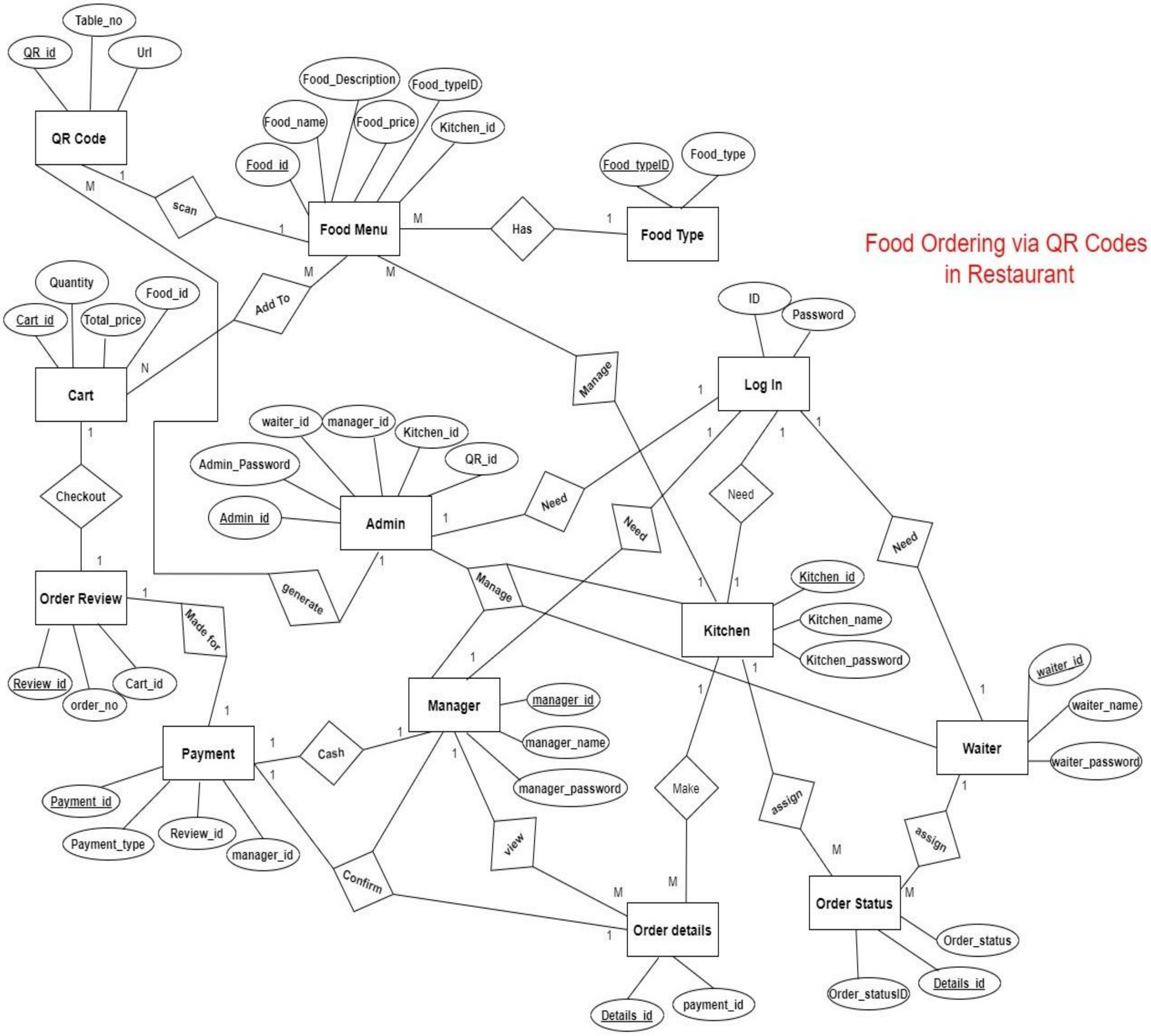


Figure 4.1: ER Diagram

4.2 Class Diagram

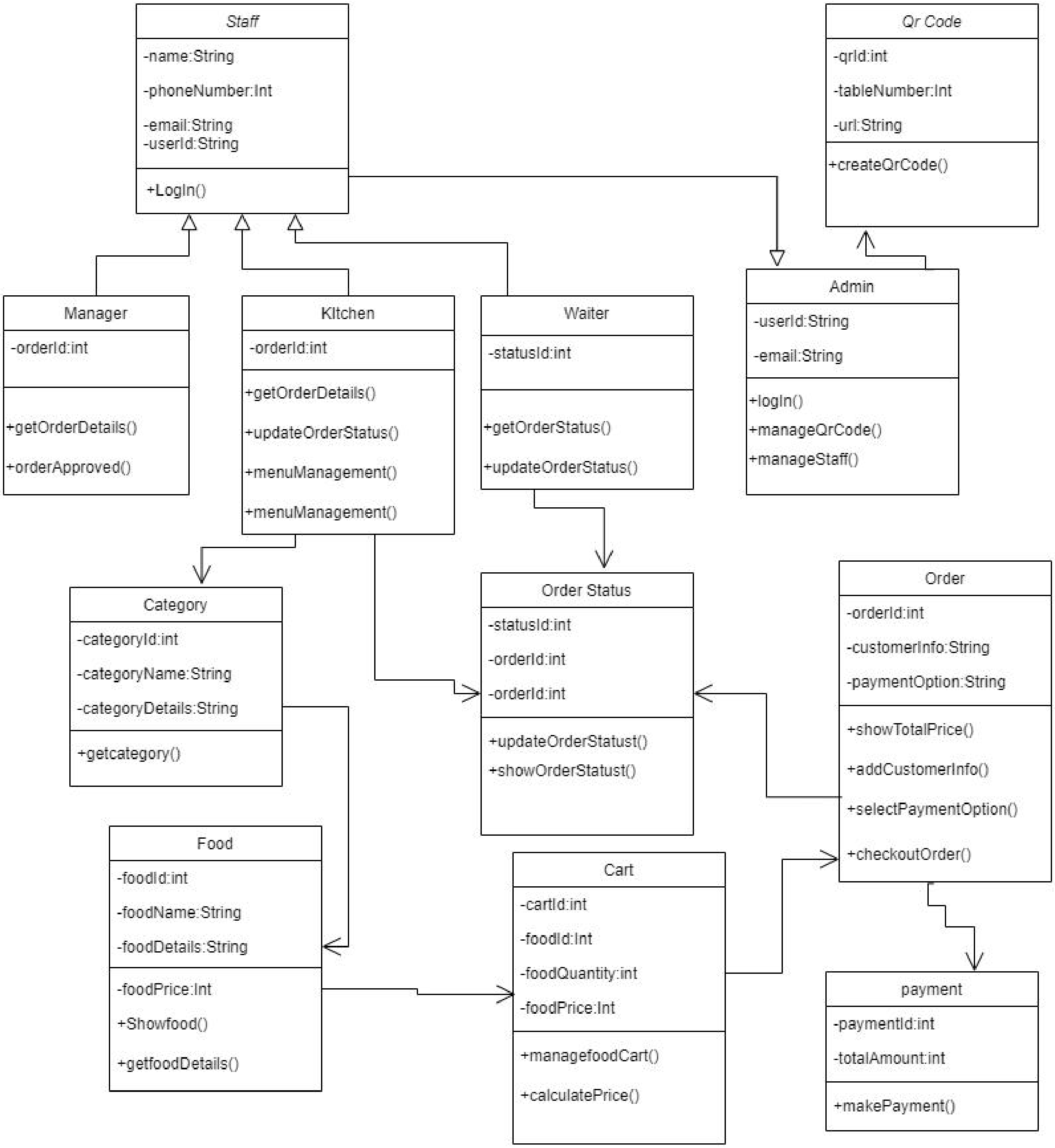


Figure 4.2: Class Diagram

4.3 Development Tools and Technology

4.3.1 User Interface Technology

- HTML, HTML5
- CSS3, CSS
- Bootstraps-4
- JavaScript, Font Awesome

4.3.2 Implementation Tools and Platforms

- Laravel framework
- Xampp
- VS Code
- WindowsCMD

CHAPTER-05
System Testing

5.1 Introduction

This Restaurant website is built for public purpose. Several types of user can use this system with many facilities. For maintaining standard quality, various kind of testing process is required. Such as-

- Functional Testing
- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

5.1.1 Features to be tested

These features to be tested:

Number	Title	Description	Priority
i	Registration Staff	If user wants to manage this site then staff has to register first.	3
ii	Login	If staff is registered then he/she can Login to see user index to access the features.	3
iii	Scan Qr code	Customer can scan qr code for viewing menu in this website	3
iv	Add to cart	Customer can add food in order cart.	3
v	Update Cart	Anyone can update their order cart.	1
vi	Payment	Users can payment through online by choosing option (bkash, dbbl) or cash.	3
vii	Update staff account	Only Admin can update staff's account.	3
viii	Logout	The session must be destroyed after Logout.	3
ix	Track order	Customer can track their order.	3

Here, 1 = Low Priority; 2 = Medium Priority; 3 = High Priority.

5.1.2 Testing Schedule

This table describe testing schedule of my project:

Test Phase	Time
Test Plan	1 Week
Test Specification	2 Week
Test Specification Team	1 Week
Component Test	3 Week
Integration Test	2 Week
System Testing	3,4 Week

Table 5.1.2: Testing Schedule

5.2 Testing Strategy

A testing strategy is a process that describes an approach to testing a software development cycle. In this process at first, I will check all the required requirements those are needed to develop my project. Then I will check that all the required requirements are properly working or not. To maintain the standard quality, I will follow some steps properly. For this I start the testing process with functional requirements testing. To maintain the quality, I did all of the tested criteria those I mentioned above.

5.3 Test Case Table

5.3.1 Test case 1 (Login)

Test Case: - 01	Test Case Name: Login
System: User Login	Subsystem: N/A
Designed By: Md. Abu Ahosan Habib	Design Date: 01-12-2021
Execute By: Md. Abu Ahosan Habib	Execution date: 02-12-2021

Step	Action	Expected System response	Pass/Fail	Comment
1	When system user fills up the user login field and click login button	If user don't enter email id then show required email id.	Pass	Email Field are required
2	When user enters emails like abc@gmail.com	The system shows this field is email required.	Pass	The valid email needs to login

Table 5.3.2: User Login

5.3.2 Test case 2 (Search Product)

Test Case: - 02	Test Case Name: Scan Qr code
System: Scan Qr code	Subsystem: N/A
Designed By: Md. Abu Ahosan Habib	Design Date: 03-12-2021
Execute By: Md. Abu Ahosan Habib	Execution date: 05-12-2021

Step	Action	Expected System response	Pass/Fail	Comment
1	When customer open Qr code scanner and scan Qr code	If qr code does not match the system then try again.	Pass	Qr code required
2	When user scan Qr code	The system shows menu.	Pass	The valid Qr code needs to application

Table 5.3.2: Search product

5.3.3 Test case 3(Track Order)

Test Case: - 03	Test Case Name: Track Order
System: Track Order	Subsystem: N/A
Designed By: Md. Abu Ahosan Habib	Design Date: 10-12-2021
Execute By: Md. Abu Ahosan Habib	Execution date: 13-12-2021

Step	Action	Expected System response	Pass/Fail	Comment
1	When system user fills up the track order field and click Track order button	If user don’t enter order id then show invalid order id	Pass	Order id required .
2	When user enters Order id like YC101010	The system shows this field is order shipping status	Pass	The valid order id needs to add problem

Table 5.3.3: Track Order

5.3.4 Test case 4(Checkout)

Test Case: - 04	Test Case Name: Checkout
System: Checkout	Subsystem: N/A
Designed By: Md. Abu Ahosan Habib	Design Date: 13-12-2021
Execute By: Md. Abu Ahosan Habib	Execution date: 16-12-202

Step	Action	Expected System response	Pass/Fail	Comment
1	When system user fills up the permission field	Cannot Checkout without select payment option.	Pass	Payment option required.
2	When user enters payment option	The system shows this field is Valid info	Pass	User order confirm.

Table 5.3.4: (Checkout)

5.4 Pass / Fail Criteria

Pass or fail criteria will be set by the test engineers. They will prepare the pass / fail criteria on the basis of which input data are worked and which do not work well. Those data that are worked well will be considered as pass criteria. And rest of the input data will be considered as fail criteria.

Now we will give the pass / fail criteria below:

- System crash will not be considered as a pass case
- If any criteria pass 100% times, then it will be considered as pass criteria only
- If data can't be displayed to the application properly, then it is also to be considered as fail criteria.

5.5 Equivalent Class Partitioning

Equivalent class partitioning is also known as equivalent partitioning. In this technique, input data is to be divided into groups. Those groups of data are expected to perform similar behavior. Each group works as same as other groups also.

The main advantage of following the approach of equivalent class partitioning is to reduce the total number of test cases from infinite to finite. Another advantage is that it can be applied to all testing levels also. For legal input value output will produce a meaning data. But for an illegal input value, the output will not produce meaningful data.

5.5.1 Black Box Testing

Black box testing is the process of test a software system without view the code. It just tests the application system using input valid and invalid. Invalid testing if the system runs link works correctly the system is incorrectly. Black box testing can be both functional and non-functional. It ignores the internal mechanism of a system. We have decided to perform the equivalent class partitioning and Boundary value analysis techniques to implement

5.5.2 White Box Testing

White box testing is also a name of testing approach which is also known as clear box testing, glass box testing, open box testing, transparent box testing, code-based testing or structural testing. It is opposite to the black box testing. In black-box testing, the internal architecture or algorithms is not known to the testers whether the whole software architecture is known to the testers while white box testing.

Testers can also predict the output of every test case for white box testing. White box testing can be classified into some levels. Such as:

- Unit Testing
- Integration Testing
- System Testing

The main advantage of white-box testing is that testing is more throughout and the testing can be started from the very beginning stage.

5.6 Testing Environment (hardware/software requirements)

Testing environment means to prepare the environment with hardware and software so that test engineers can be able to execute test cases as required. Besides hardware and software usage, network configuration might be needed to execute test plans.

1. Browser: -Google Chrome, Firefox.
2. Core i5, Ram# 8GB, SSD-120GB, HDD-1TB.

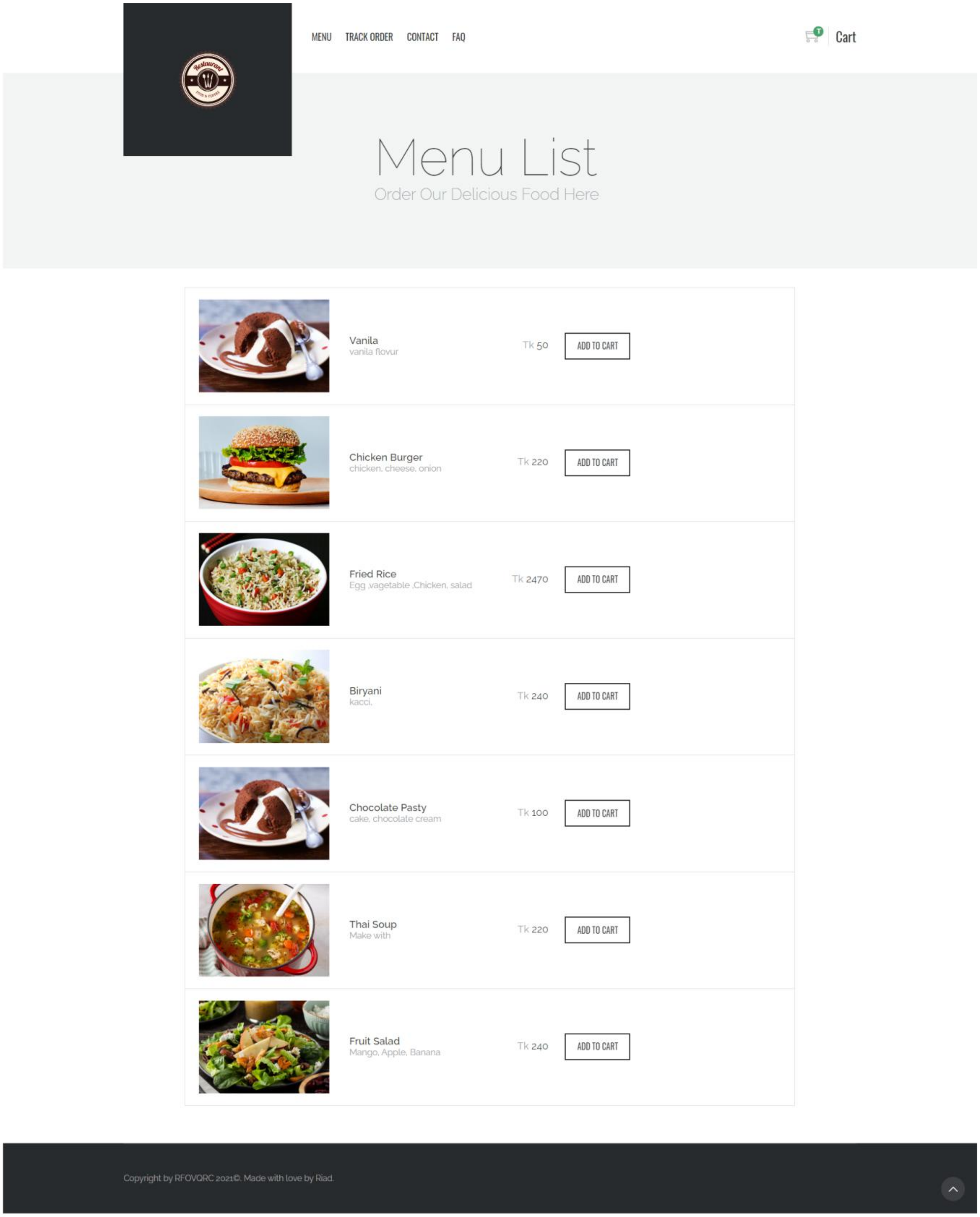
CHAPTER – 06

User Interface

6.1 User Interface(Customer)

1. Scan QR Code

2. After Scaning qr code show menu page



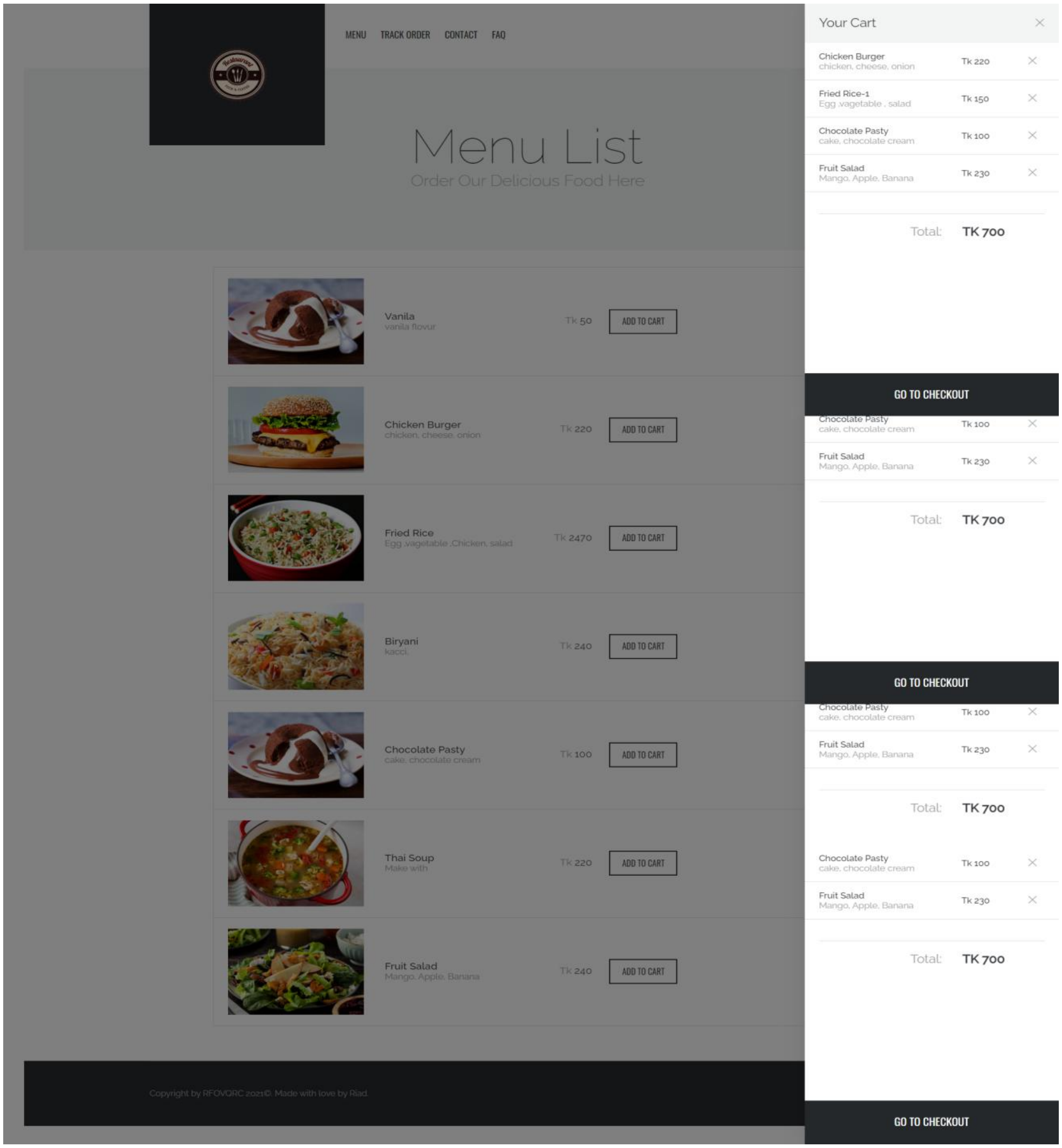
Menu List

Order Our Delicious Food Here

Vanila vanila flovur	Tk 50	ADD TO CART
Chicken Burger chicken, cheese, onion	Tk 220	ADD TO CART
Fried Rice Egg ,vagetable ,Chicken, salad	Tk 2470	ADD TO CART
Biryani kacci,	Tk 240	ADD TO CART
Chocolate Pasty cake, chocolate cream	Tk 100	ADD TO CART
Thai Soup Make with	Tk 220	ADD TO CART
Fruit Salad Mango, Apple, Banana	Tk 240	ADD TO CART

Copyright by RFOVQRC 2021©. Made with love by Riad.

3. Add to cart



MENU TRACK ORDER CONTACT FAQ

Menu List

Order Our Delicious Food Here

Vanila vanila flovur	Tk 50	ADD TO CART
Chicken Burger chicken, cheese, onion	Tk 220	ADD TO CART
Fried Rice Egg ,vagetable ,Chicken, salad	Tk 2470	ADD TO CART
Biryani kacci.	Tk 240	ADD TO CART
Chocolate Pasty cake, chocolate cream	Tk 100	ADD TO CART
Thai Soup Make with	Tk 220	ADD TO CART
Fruit Salad Mango, Apple, Banana	Tk 240	ADD TO CART

Copyright by RFOVQRC 2021©. Made with love by Riad.

Your Cart ×

Chicken Burger chicken, cheese, onion	Tk 220	×
Fried Rice-1 Egg ,vagetable , salad	Tk 150	×
Chocolate Pasty cake, chocolate cream	Tk 100	×
Fruit Salad Mango, Apple, Banana	Tk 230	×

Total: TK 700

GO TO CHECKOUT

Chocolate Pasty cake, chocolate cream	Tk 100	×
Fruit Salad Mango, Apple, Banana	Tk 230	×

Total: TK 700

GO TO CHECKOUT

Chocolate Pasty cake, chocolate cream	Tk 100	×
Fruit Salad Mango, Apple, Banana	Tk 230	×

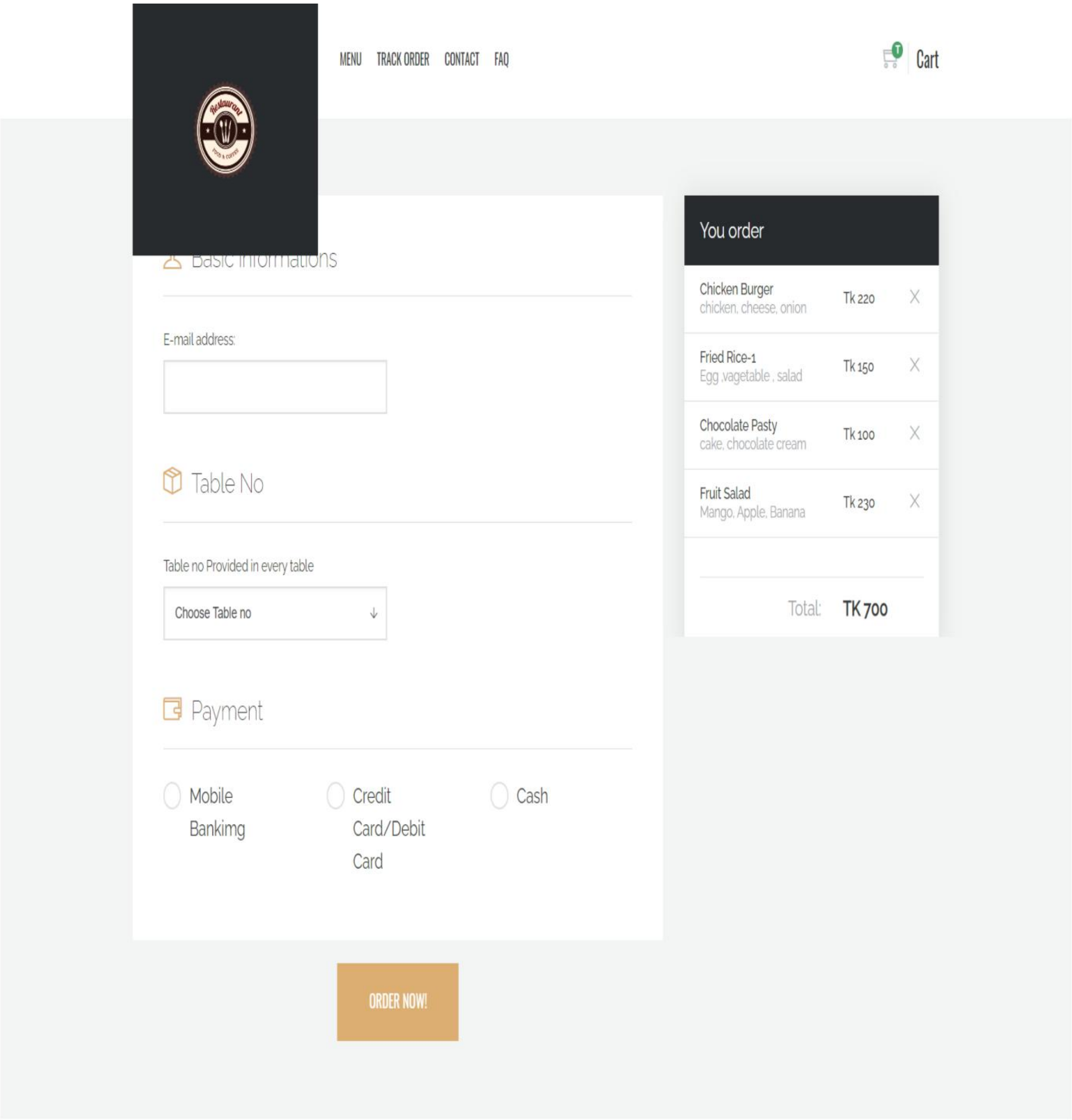
Total: TK 700

Chocolate Pasty cake, chocolate cream	Tk 100	×
Fruit Salad Mango, Apple, Banana	Tk 230	×

Total: TK 700

GO TO CHECKOUT

4. Check Out



Basic Informations

E-mail address:

Table No

Table no Provided in every table

Choose Table no

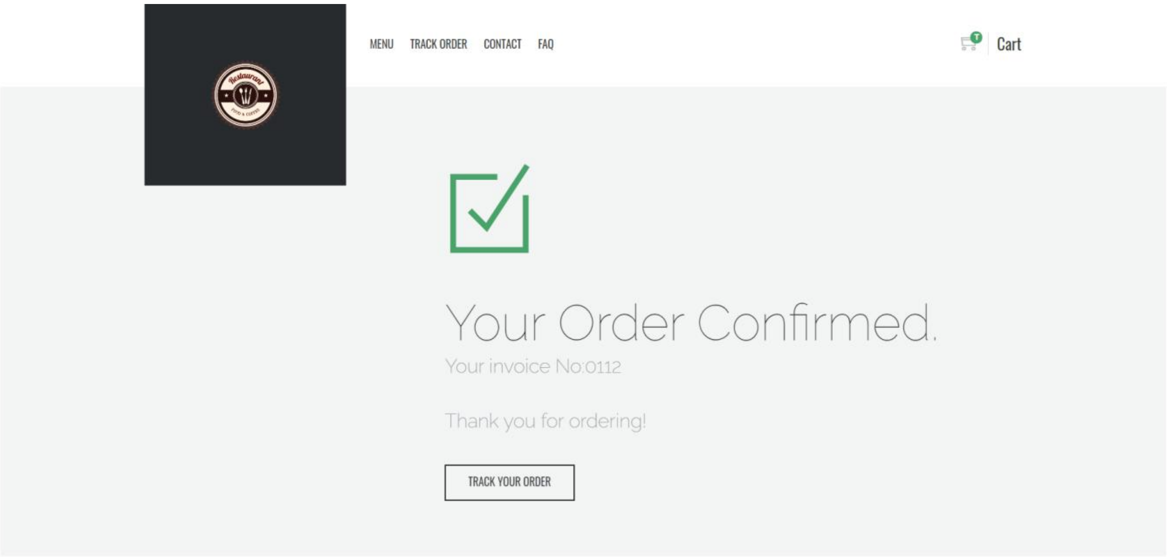
Payment

○ Mobile Bankimg ○ Credit Card/Debit Card ○ Cash

You order

Chicken Burger chicken, cheese, onion	Tk 220	X
Fried Rice-1 Egg ,vagetable , salad	Tk 150	X
Chocolate Pasty cake, chocolate cream	Tk 100	X
Fruit Salad Mango, Apple, Banana	Tk 230	X
Total:	TK 700	

5. Confirm Order

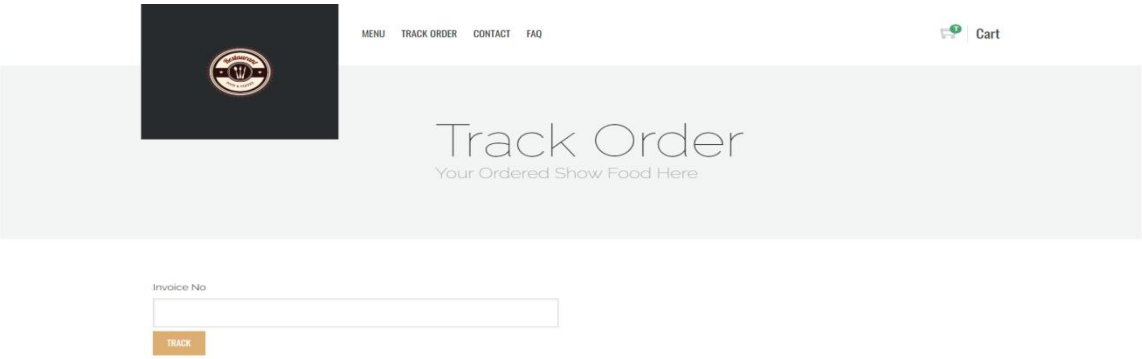


Your Order Confirmed.

Your invoice No:0112

Thank you for ordering!

6. Order Details Show after Searching

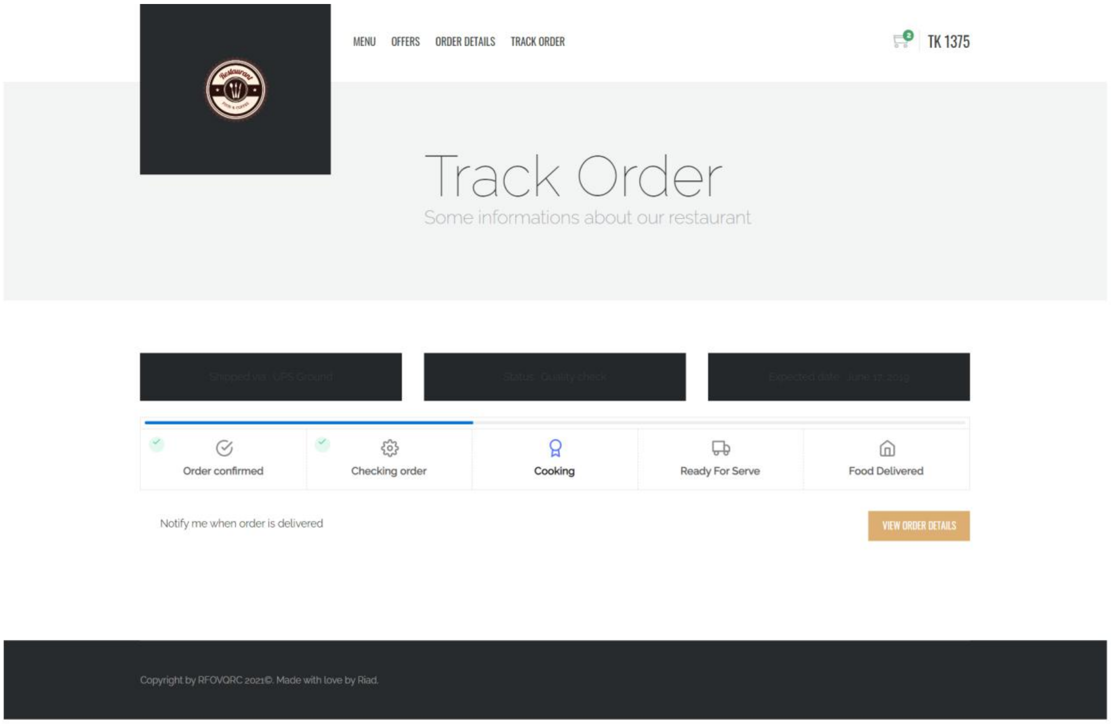


Track Order

Your Ordered Show Food Here

Invoice No

7. Track Order



Track Order

Some informations about our restaurant

Shipped via : UPS Ground

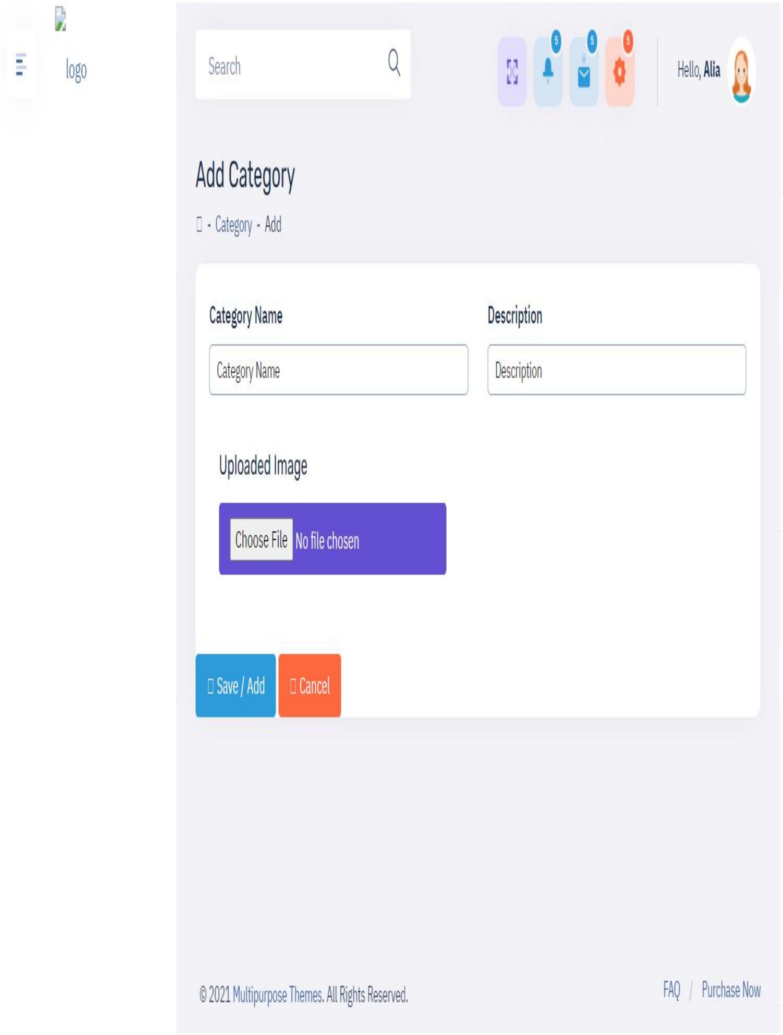
Status : Quality check

Expected date : June 17, 2019

Notify me when order is delivered

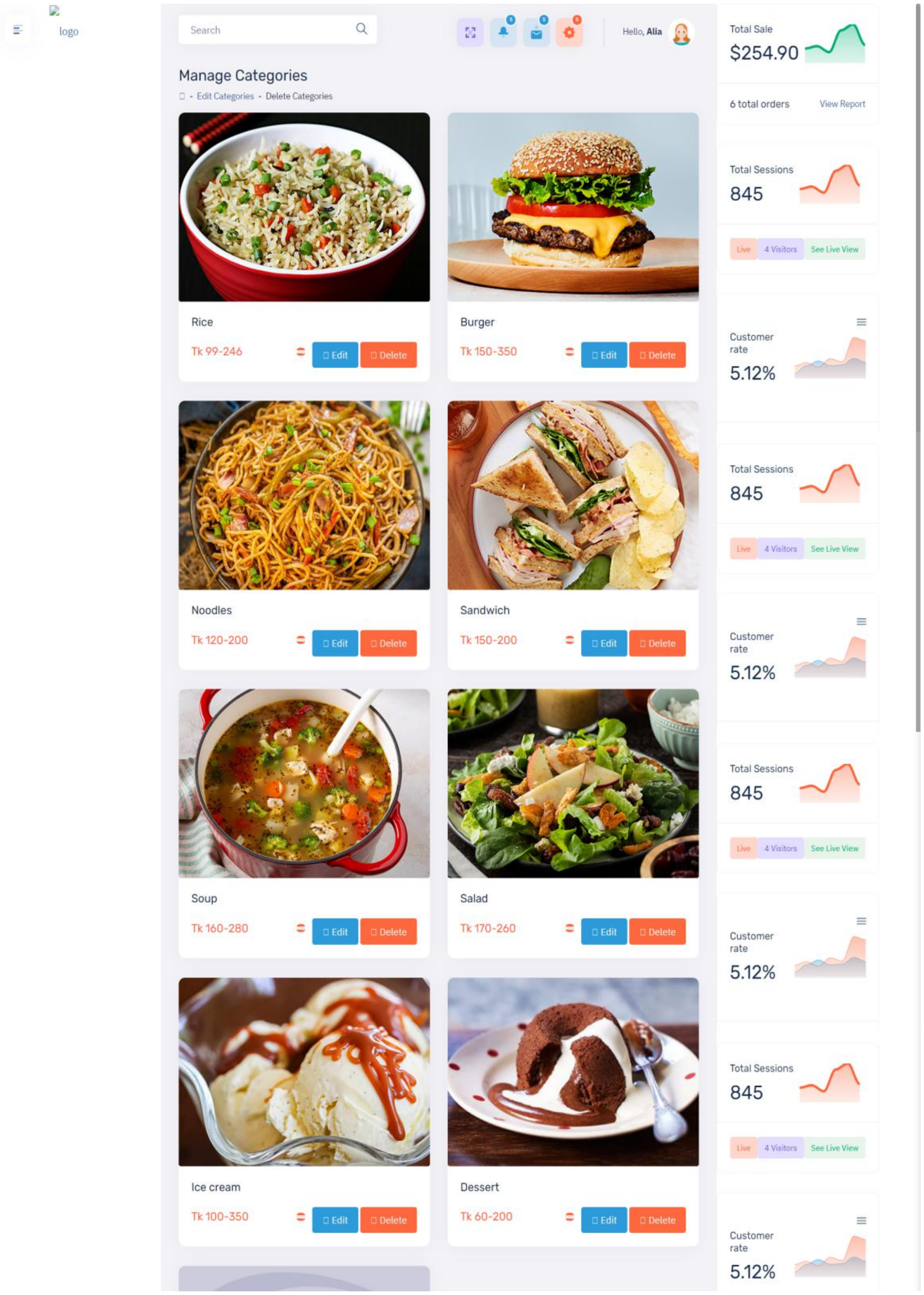
Copyright by RFOVQRC 2021©. Made with love by Riad.

8. Admin Add Category



© 2021 Multipurpose Themes. All Rights Reserved.

9. Manage Category



Manage Categories

□ - Edit Categories - Delete Categories

Rice
Tk 99-246

Burger
Tk 150-350

Noodles
Tk 120-200

Sandwich
Tk 150-200

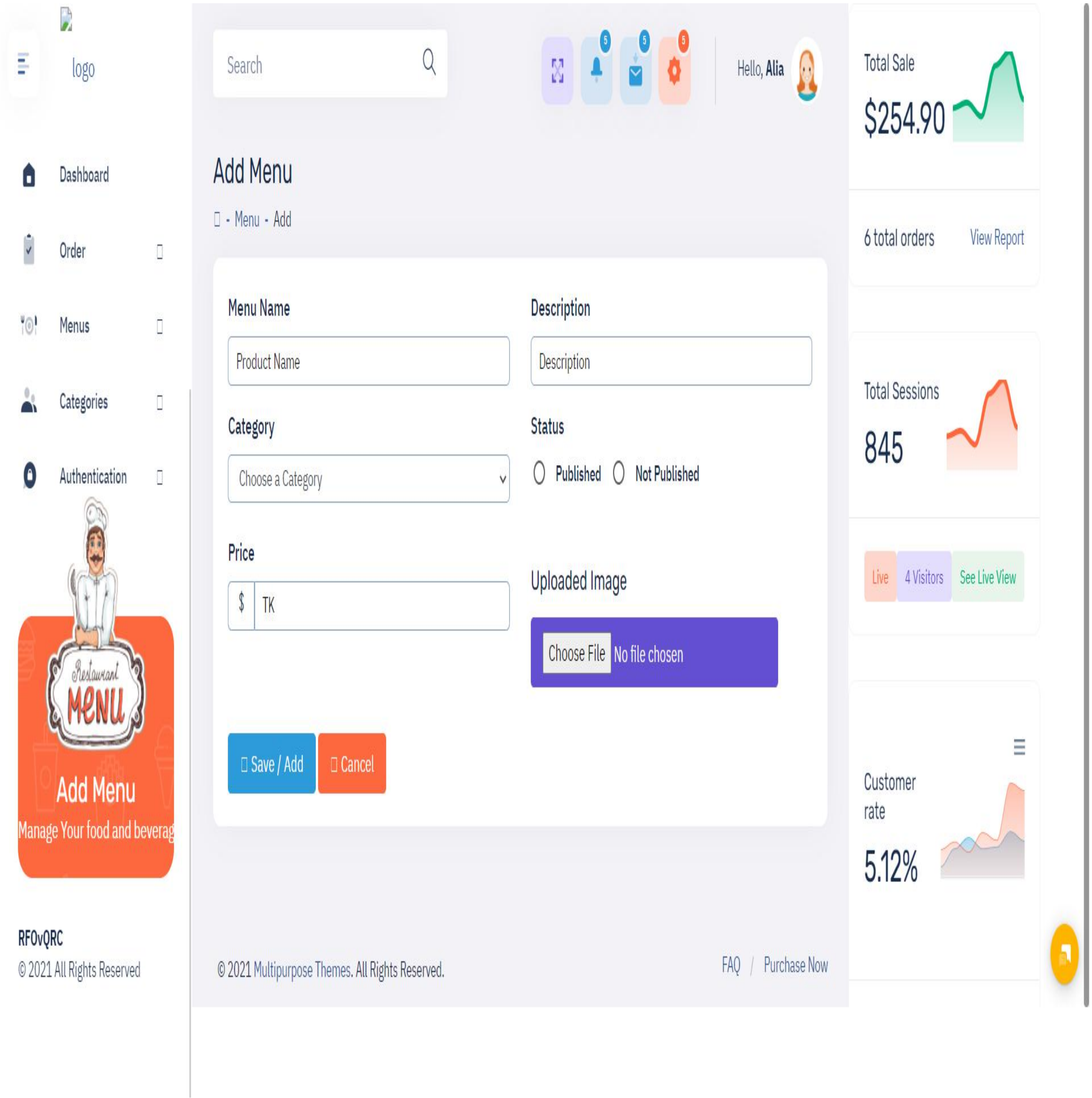
Soup
Tk 160-280

Salad
Tk 170-260

Ice cream
Tk 100-350

Dessert
Tk 60-200

10. Add Menu



Add Menu

Menu - Add

Menu Name: Product Name

Description: Description

Category: Choose a Category

Status: ○ Published ○ Not Published

Price: $ TK

Uploaded Image: Choose File No file chosen

Total Sale $254.90

6 total orders View Report

Total Sessions 845

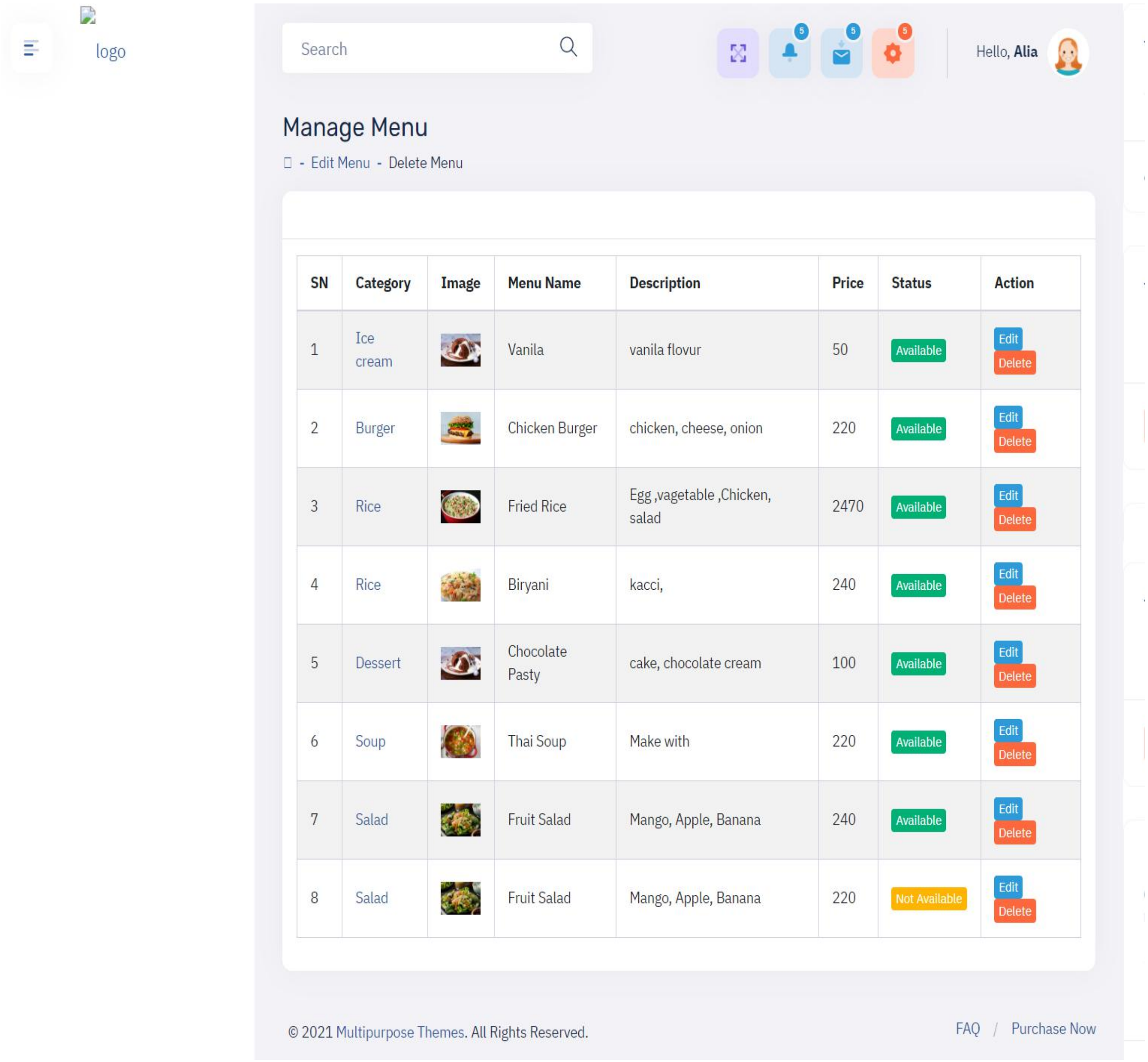
Live 4 Visitors See Live View

Customer rate 5.12%

RFOvQRC
© 2021 All Rights Reserved

© 2021 Multipurpose Themes. All Rights Reserved.

11. Manage Menu



Manage Menu

- Edit Menu - Delete Menu

SN	Category	Image	Menu Name	Description	Price	Status	Action
1	Ice cream		Vanila	vanila flovur	50	Available	Edit Delete
2	Burger		Chicken Burger	chicken, cheese, onion	220	Available	Edit Delete
3	Rice		Fried Rice	Egg ,vagetable ,Chicken, salad	2470	Available	Edit Delete
4	Rice		Biryani	kacci,	240	Available	Edit Delete
5	Dessert		Chocolate Pasty	cake, chocolate cream	100	Available	Edit Delete
6	Soup		Thai Soup	Make with	220	Available	Edit Delete
7	Salad		Fruit Salad	Mango, Apple, Banana	240	Available	Edit Delete
8	Salad		Fruit Salad	Mango, Apple, Banana	220	Not Available	Edit Delete

© 2021 Multipurpose Themes. All Rights Reserved.

FAQ / Purchase Now

CHAPTER-07
Project Summary

7.1 Limitations

- The main limitation is system cannot handle million data and signals at time.
- There is no registration system included in this application for recruiters; recruiters have to view the talents profile without registration.

7.2 Obstacles and Achievement

Obstacles:

- Learning new technology and new environment is a big issue.
- It's very difficult to complete a work within limited time.
- To collect requirements is a very tough.

Achievement:

- Successfully built a project.
- Learn a new technology.
- Deploy a project is a new experience.
- Learned the real-life experience by uploading project on the live server.
- Know about document and the development process.

7.3 Future Work

Though the system was developed as much as needed and its work properly. But I have to add something new features to make the systems fulfillment. The future work will include some major changes, such as-

- Real payment system.
- Live menu status: which food is now available or not.
- Send order status notification to the customer

REFERENCES

1. How To: Write a Project proposal (Online) URL: https://mogadalai.wordpress.com/2007/05/28/howto-write-a-project-proposal/

2. What is QR Code. Retrieved from https://www.qrcode.com/en/about/

3. QR code model 1 model 2. Retrieved from https://www.qrcode.com/en/codes/model12.html/

4. https://www.w3schools.com/

5. https://www.umpir/ump.edu.my/id/eprint/12526/

6. "Implementing Customizable Online Food Ordering System Using Web Based Application" http://ijiset.com/vol2/v2s4/IJISET_V2_I4_112.pdf

7. "Implementation of Paperless Dining Experience using Client-Server Model with Cloud and QR Technology" http://www.erpublications.com/sitepanel/download.php?filename=download/download_25_01_2015_12_20_20.pdf

8. Uber Eats: https://www.ubereats.com/